The complexity of optimal design of temporally connected graphs

Eleni C. Akrida[1], Leszek Gąsieniec[1], George B. Mertzios[2], and Paul G. Spirakis[1,3]

[1] Department of Computer Science, University of Liverpool, UK
{Eleni.Akrida2,L.A.Gasieniec,P.Spirakis}@liverpool.ac.uk
[2] School of Engineering and Computing Sciences, Durham University, UK
George.Mertzios@durham.ac.uk
[3] Computer Technology Institute & Press "Diophantus" (CTI), Patras, Greece

Abstract. We study the design of small cost temporally connected graphs, under various constraints. We mainly consider undirected graphs of n vertices, where each edge has an associated set of discrete availability instances (labels). A journey from vertex u to vertex v is a path from u to v where successive path edges have strictly increasing labels. A graph is temporally connected iff there is a (u, v)-journey for any pair of vertices $u, v,\ u \neq v$. We first give a simple polynomial-time algorithm to check whether a given temporal graph is temporally connected. We then consider the case in which a designer of temporal graphs can *freely choose* availability instances for all edges and aims for temporal connectivity with very small *cost*; the cost is the total number of availability instances used. We achieve this via a simple polynomial-time procedure which derives designs of cost linear in n. We also show that the above procedure is (almost) optimal when the underlying graph is a tree, by proving a lower bound on the cost for any tree. However, there are pragmatic cases where one is not free to design a temporally connected graph anew, but is instead *given* a temporal graph design with the claim that it is temporally connected, and wishes to make it more cost-efficient by removing labels without destroying temporal connectivity (redundant labels). Our main technical result is that computing the maximum number of redundant labels is APX-hard, i.e., there is no PTAS unless $P = NP$. On the positive side, we show that in dense graphs with random edge availabilities, there is asymptotically almost surely a very large number of redundant labels. A temporal design may, however, be *minimal*, i.e., no redundant labels exist. We show the existence of minimal temporal designs with at least $n \log n$ labels.

1 Introduction and motivation

A temporal network is, roughly speaking, a network that changes with time. A great variety of modern and traditional networks are not static and change over time. For example, social networks, wired or wireless networks may change dynamically, transport network connections may only operate at certain times, etc. Dynamic networks in general have been attracting attention over the past years [6, 9, 10, 13, 27], exactly because they model real-life applications. In this work, following the model of [19,25] and [1], we consider *discrete time* and restrict our attention to systems in which only the connections between the participating entities may change but the entities remain unchanged. So we consider networks, the links of which are available only at certain discrete time instances, e.g. days or hours. This is a natural assumption when the dynamicity of the system is inherently discrete, e.g., in synchronous mobile distributed systems that operate in discrete rounds. Moreover, it gives a purely combinatorial flavour to the resulting models and problems.

In several such dynamic settings, maintaining connections may come at a cost; consider the transport network example above or an unstable chemical or physical structure, where energy is required to keep a link available. We define the cost as the total number of discrete time instances at which the network links become available. We focus on design issues of temporal networks that are temporally connected; a temporal network is temporally connected if information can travel over time from any node to any other node following *journeys*, i.e., paths whose successive edges have strictly increasing availability time instances. If one has absolute freedom to design a small cost temporally connected temporal network on an underlying static network, i.e, choose the edge availabilities, then a reasonable design would be to select a rooted spanning tree and choose appropriate availabilities to construct time-respecting paths from the leaves to the root and *then* from the root back to the leaves. However, in more complicated scenarios one may not be free to *choose* edge availabilities arbitrarily but instead *specific* link availabilities might pre-exist for the network; then, one is able to design a temporally connected temporal network using only the pre-existing availabilities or a subset of them. Imagine a hostile network on a complete graph where availability of a link means a break in its security, e.g., when the guards change shifts, and only then are we able to pass a message through the link. So, if we wish to send information through the network, we may only use the times when the shifts change and it is reasonable to try and do so by using as few of these breaks as possible. In such scenarios, we may need to first verify that the pre-existing edge availabilities indeed define a temporally connected temporal network. Then, we may try to reduce the cost of the design by *removing* unnecessary (redundant) edge availabilities if possible, without losing temporal connectivity. Consider, again, the clique network of n vertices with one time availability per edge; it is clearly temporally connected with cost $\Theta(n^2)$. However, it is not straightforward if all these edge availabilities are necessary for temporal connectivity. We resolve here the complexity of finding the maximum number of redundant labels in any given temporal graph.

1.1 The model and definitions

It is generally accepted to describe a network topology using a graph, the vertices and edges of which represent the communicating entities and the communication opportunities between them respectively. We consider graphs whose edge availabilities are described by sets of positive integers (labels), one set per edge.

Definition 1 (Temporal Graph). *Let $G = (V, E)$ be a (di)graph. A temporal graph on G is an ordered triple $G(L) = (V, E, L)$, where $L = \{L_e \subseteq \mathbb{N}^* : e \in E\}$ is an* assignment *of labels to the edges (arcs) of G. L is called a* labelling *of G.*

Definition 2 (Time edge). *Let $e = \{u, v\}$ (resp. $e = (u, v)$) be an edge (resp. arc) of the underlying (di)graph of a temporal graph and consider a label $l \in L_e$. The ordered triplet (u, v, l) is called* time edge*.*

Note that an undirected edge $e = \{u, v\}$ is associated with $2 \cdot |L_e|$ time edges, namely both (u, v, l) and (v, u, l) for every $l \in L_e$.

The labels of an edge (arc) e are the *discrete time instances* at which e is available. In many networks and in several applications, the availability of links comes at a cost. For example, in secure networks there is a cost (per discrete time instance) to keep a link secure. We abstract such considerations by the concept of the *cost* of a temporal graph and wish to have temporal graphs of low cost.

Definition 3 (Cost of a labelling). *Let $G(L) = (V, E, L)$ be a temporal (di)graph and L be its labelling. The* cost *of L is defined as $c(L) = \sum_{e \in E} |L_e|$.*

A basic assumption that we follow here is that when a message or an entity passes through an available link at time t, then it can pass through a subsequent link only at some time $t' > t$ and only at a time at which that link is available.

Definition 4 (Journey). *A* temporal path *or* journey *j from a vertex u to a vertex v ((u, v)-journey) is a sequence of time edges (u, u_1, l_1), (u_1, u_2, l_2), ... , (u_{k-1}, v, l_k), such that $l_i < l_{i+1}$, for each $1 \leq i \leq k - 1$. We call the last time label, l_k,* a*rrival time of the journey.*

Definition 5 (Foremost journey). *A (u, v)-journey j in a temporal graph is called* foremost journey *if its arrival time is the minimum arrival time of all (u, v)-journeys' arrival times, under the labels assigned to the underlying graph's edges. We call this arrival time the* temporal distance*, $\delta(u, v)$, of v from u.*

In this work, we focus on *temporally connected* temporal graphs, i.e., temporal graphs that have the following property:

Definition 6 (Property TC). *A temporal (di)graph $G(L) = (V, E, L)$ satisfies the property TC, or equivalently L satisfies TC on G, if for any pair of vertices $u, v \in V$, $u \neq v$, there is a (u, v)-journey* and *a (v, u)-journey in $G(L)$. A temporal (di)graph that satisfies the property TC is called* temporally connected*.*

Example *An undirected complete graph, K_n, is temporally connected under any labelling L with $L_e \neq \emptyset$ for every $e \in E(K_n)$. Indeed, there is a (u, v)-journey and a (v, u)-journey between any $u, v \in V(K_n)$, $u \neq v$, namely the time edge (u, v, l) and the time edge (v, u, l) respectively, for any $l \in L_{\{u,v\}}$.*

Definition 7 (Minimal temporal graph). *A temporal graph* $G(L) = (V, E, L)$ *over a (strongly) connected (di)graph is* minimal *if* $G(L)$ *has the property TC, and the removal of any label from any* L_e, $e \in E$, *results in a* $G(L')$ *that* does not *have the property TC.*

Definition 8 (Removal profit). *Let* $G(L) = (V, E, L)$ *be a temporally connected temporal graph. The* removal profit $r(G, L)$ *is the largest total number of labels that can be removed from* L *without violating TC on* G.

Here, removal of a label l from L refers to the removal of l only from a particular edge and not from all edges that are assigned label l, i.e., if $l \in L_{e_1} \cap L_{e_2}$ and we remove l from both L_{e_1} and L_{e_2}, it counts as two labels removed from L.

Notice that if many edges have the same label, we can encounter *trivial cases* of minimal temporal graphs. For example, the complete graph where every edge appears at time, say $t = 5$, is minimal but there are no journeys of length larger than 1. To avoid cases where minimality is caused merely due to the assignment of the same label(s) to many (or all) edges, we will often consider a special sub-category of (single-labelled) temporal graphs:

Definition 9 (SLSE temporal graphs). *A Single-label-single-edge (SLSE) temporal graph is a temporal graph, each edge of which has a single label and no two edges have the same label, i.e., each label is assigned to (at most) a single edge. A labelling that gives an SLSE temporal graph is also called* SLSE labelling.

1.2 Previous work and our contribution

In recent years, there is a growing interest in distributed computing systems that are inherently dynamic. For example, temporal dynamics of network flow problems were considered in a set of pioneering papers [14, 15, 20, 21]. The model we consider here is very closely related to the single-labelled model of the seminal paper of [19] as well as the multi-labelled model of [25]. In [19], the authors consider the case of one *real* label per edge and examine how basic graph properties change when we impose the temporal condition; here, we extend that model by considering multiple labels per edge but we restrict our focus to integer labels. In [25], the model of [19] is also extended to many labels per edge and the authors mainly examine the number of labels needed for a temporal design of a network to guarantee several graph properties with certainty. The latter also defined the cost notion and, amongst other results, gave an algorithm to compute foremost journeys which can be used to decide property TC. However, the time complexity of that algorithm was pseudo-polynomial, as it was dominated by the cube of the maximum label used in the given labelling.

In fact, the problem of testing whether a dynamic graph is temporally connected has been studied before in various settings [7, 9, 33]. The authors of [9] propose an algorithm for computing foremost journeys in a model of evolving graphs, where nodes and edges are associated with lists of time intervals, representing their existence over time, and each edge has a traversal time. In a similar setting, [33] studies temporal reachability graphs, in which a (u, v)-edge is present at time t if (in the corresponding time-varying graph) there is a (u, v)-journey leaving u after t and arriving at v after at most some specified time-interval. In [7], the authors investigate discrete-time evolving graphs, for which they compute the *transitive closure of journeys*, i.e., a static directed graph whose edges represent potential journeys. The algorithm they propose depends on the maximum label used, the number of vertices, and the maximum number of edges that simultaneously exist.

Here, we show that if the designer of a temporal graph can select edge availabilities freely, then an asymptotically optimal linear-cost (in the size of the graph) design that satisfies TC can be easily obtained (cf. Section 3). We give a matching lower bound to indicate optimality, in the case where the underlying graph is a tree. However, there are pragmatic cases where one is not free to design a temporal graph anew; instead, one is *given* a set of possible availabilities per edge with the claim that they satisfy TC and the constraint that they may only use them or a subset of them for their design. We also propose a simple algorithm to verify TC in low polynomial time (cf. Section 2). The *given* design may also be minimal; we partially characterise minimal designs in Section 4. On the other hand, there may be some labels of the initial design that can be removed without violating TC (and also result in a lower cost). In this case, how many labels can we remove at best? Our main technical result is that this problem is APX-hard, i.e. it has no PTAS unless $P = NP$. On the positive side, we show that in the case of complete graphs and random graphs, if the labels are also assigned at random, there is aymptotically almost surely a very large number of labels that can be removed without violating TC. A preliminary version of this work appeared in the 13^{th} Workshop on Approximation and Online Algorithms, WAOA 2015 [2].

Stochastic aspects and/or survivability of network design were also considered in [17, 23, 24].

Further related work Below, we provide a short survey of papers with studies on networks labelled by time units or segments, in addition to the ones mentioned above.

Labelled Graphs. Labelled graphs have been widely used both in Computer Science and in Mathematics, e.g., [29].

Continuous Availabilities (Intervals). Some authors have assumed the availability of an edge for a whole time-interval $[t_1, t_2]$ or multiple such time-intervals and not just for discrete moments as we assume here. Examples of such studies are [3, 9, 15].

Dynamic Distributed Networks. In recent years, there is a growing interest in distributed computing systems that are inherently dynamic [5, 6, 10, 12, 13, 22, 26, 27, 30, 32].

Distance labelling. A distance labelling of a graph G is an assignment of unique labels to vertices of G so that the distance between any two vertices can be inferred from their labels alone [16, 18].

Random labellings. Random temporal networks have been considered before, e.g., in [1, 11, 12]. In [11], the authors model opportunistic mobile networks as a type of random temporal networks, where each edge exists at each time-step with a fixed probability, and show a small diameter in general for that type of networks. In [12], the authors examine the speed of information dissemination in a type of dynamic graphs, where each edge exists at each time-step with some probability depending on whether it existed in the previous time-step. The *Expected Temporal Diameter* of the model of (random) temporal graphs that we consider here was first examined in [1].

2 A low polynomial time algorithm for deciding TC

In this section, we propose a simple polynomial-time algorithm which, given a temporal (di)graph $G(L) = (V, E, L)$ and a source vertex $s \in V$, computes a *foremost* (s, v)-journey, for every $v \neq s$, if such a journey exists. We conjecture that our algorithm is optimal.

Theorem 1. *Algorithm 1 satisfies the following, for every vertex $v \in V,\ v \neq s$:*

(a) If $arrival\_time[v] < +\infty$, then there exists a foremost journey from s to v, the arrival time of which is exactly $arrival\_time[v]$. This journey can be constructed by following the $parent[v]$ pointers in reverse order.
(b) If $arrival\_time[v] = +\infty$, then no (s, v)-journey exists.
(c) The time complexity of Algorithm 1 is dominated by the sorting time of the set of time edges.

Proof sketch. The algorithm actually considers each existing label in the sequence of time labels, from the smallest to the largest one. For each label considered, it computes the foremost journeys from s which arrive at that time[4]. The algorithm examines each time edge exactly once. □

Corollary 1. *The time complexity of Algorithm 1 is $O\big(c(L) \cdot \log c(L)\big)$.*

Proof. The time complexity of the algorithm is dominated by the sorting time of $S(L)$. One can sort $S(L)$ by comparison-based sorting resulting in running time $O(|S(L)| \cdot \log |S(L)|) = O\big(c(L) \cdot \log c(L)\big)$. □

Conjecture *We conjecture that any algorithm that computes journeys out of a vertex s must sort the time edges by their labels, i.e., we conjecture that Algorithm 1 is asymptotically optimal with respect to the running time.*

Note that Algorithm 1 can even compute foremost (s, v)-journeys, if they exist, that *start* from a given time $t_{start} > 0$. Simply, one ignores the time edges with labels smaller than the start time.

[4] One can prove this by induction.

Algorithm 1 Foremost journey algorithm

Input: A temporal (di)graph $G(L) = (V, E, L)$ of n vertices, the set of all time edges of which is denoted by $S(L)$; a designated source vertex $s \in V$

Output: A foremost (s, v)-journey from s to all $v \in V \setminus \{s\}$, where such a journey exists; if no (s, v)-journey exists, then the algorithm reports it.

Sort $S(L)$ in increasing order of labels; // Note that $|S(L)| = c(L)$ if G is directed, and $|S(L)| = 2c(L)$ if G is undirected.
Let S' be the sorted array of time edges according to time labels;
$R := \{s\}$; // The set of vertices to which s has a foremost journey
$arrival\_time[s] := 0$;
for all $v \in V \setminus \{s\}$ **do**
    $parent[v] := \emptyset$;
    $arrival\_time[v] := +\infty$;
for all time edges (a, b, l) in the order given by S' **do**
    if $a \in R$ **and** $b \notin R$ **and** $arrival\_time[a] < l$ **then**
        $parent[b] := a$;
        $arrival\_time[b] := l$;
        $R := R \cup \{b\}$;

3 Asymptotically cost-optimal design for TC in undirected graphs.

In this section, we study temporal design issues on connected undirected graphs, so that the resulting temporal graphs are temporally connected. In this scenario, the designer has absolute freedom to choose the edge availabilities of the underlying graph.

Lemma 1. *There is an infinite family of graphs G_n of n vertices, for which the cost of any labelling that satisfies TC is at least $2n - 3$.*

Proof. Consider the star graph of n vertices, $n \geq 3$. Let v_n be the root and $v_1, v_2, \ldots, v_{n-1}$ be the leaves. In any labelling on the star graph, which assigns only one label to two (or more) edges (v_n, v_x), (v_n, v_y), $x, y = 1, 2, \ldots, n-1$, $x \neq y$, at least one of the vertices v_x, v_y cannot reach the other via a journey. Therefore, any TC satisfying labelling on the star graph must assign at least 2 labels to all edges of the graph, except possibly on one edge where it assigns a single label. The TC satisfying labelling which assigns labels $1, 3$ to all edges except for one and label 2 to the remaining edge has, therefore, minimum cost, namely $2n - 3$ (cf. Figure 1).

□

In fact, the result of Lemma 1 is optimal for any tree; it is indeed strictly contained in Theorem 2(a), but the proof of Lemma 1 is significantly simpler. Theorem 2 shows a lower bound for trees and an asymptotically optimal[5] way of labelling any connected undirected graph to satisfy TC.

[5] Any connected undirected graph needs at least $n - 1$ labels on its edges to be temporally connected, and we show a TC satisfying labelling of $2n - 3 = \Theta(n)$ labels.

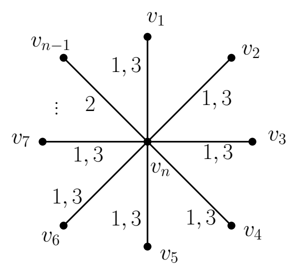

Fig. 1. Labelling a star graph in an optimal way

Theorem 2. *(a) For any tree $G = (V, E)$ of n vertices and for any labelling L that satisfies the property TC on G, the cost of L is $c(L) \geq 2n - 3$.*

(b) Given a connected undirected graph $G = (V, E)$ of $n \geq 2$ vertices, we can design a labelling L of cost $c(L) = 2n - 3$ that satisfies the property TC on G. L can be computed in polynomial time.

Proof.

(a) We prove the statement by induction on the number of vertices of the tree.

Base Case. It is easy to see that the statement holds for any tree of $n \leq 4$ vertices.

Induction Hypothesis. Assume that at least $2n-3$ labels are *necessary* to satisfy TC on any tree of $n \leq k$ vertices, $k \in \mathbb{N}$.

Inductive Step. We will show that at least $2(k+1) - 3 = 2k - 1$ labels are necessary to satisfy TC on any tree of $k + 1$ vertices.

Let $G = (V, E)$ be an arbitrary tree of $k + 1$ vertices and let L be an arbitrary labelling of G that satisfies TC on G. Consider a leaf, $u \in V$, of G and its unique neighbour, $u' \in V$. Note that L must assign at least one label to the edge $\{u, u'\}$ to "enable" a journey between them. Now, let L' be the sub-labelling of L on $G \setminus u$. First, we show that, for L to satisfy TC on G, it must be that L' satisfies TC on $G \setminus u$.

Assume, to the contrary, that L' does not satisfy TC on $G \setminus u$. Then, there exist two vertices $x, x' \in V(G \setminus u)$ such that the only journey(s) from x to x' in $G(L)$ go through u; let J be a (x, x')-journey in $G(L)$. It must be:

$$J = \big((x, v_1, l_0), \ldots, (v_z, u', l_z), (u', u, l_{small}),\\ (u, u', l_{big}), (u', v_{z'}, l_{z'}), \ldots, (v_{last}, x', l_{last})\big),$$

for some $v_0, \ldots, v_{last}$ and $l_0 < \ldots < l_z < l_{small} < l_{big} < l_{z'} < \ldots < l_{last}$ (cf. Figure 2).

But, then the sub-journey of J which "ignores" the time-edges $(u', u, l_{small}), (u, u', l_{big})$ is still a (x, x')-journey in $G(L)$, which contradicts the fact that all (x, x')-journeys in $G(L)$ go through u. Therefore, L' must satisfy TC on $G \setminus u$. Since $G \setminus u$ is a tree of k vertices itself, it must be that $c(L') \geq 2k - 3$ (by Induction Hypothesis).

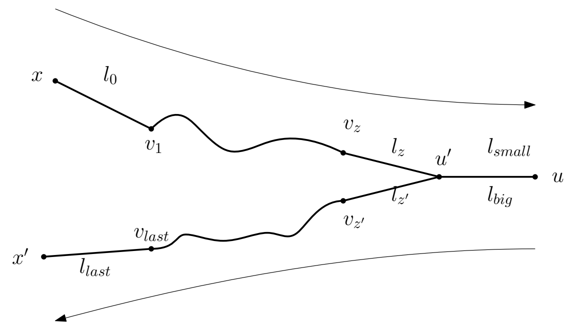

Fig. 2. A (x, x')-journey going through u.

If $c(L') \geq 2k - 2$, then (since L assigns at least one label to the edge $\{u, u'\}$), we have $c(L) \geq 2k - 2 + 1 = 2k - 1$ and the Theorem holds.
It remains to check the case where $c(L') = 2k - 3$ and L' satisfies TC on $G \setminus u$. L' must assign at least one label to every edge of $G \setminus u$ to satisfy TC on it. Also, it must assign exactly one label to at least one edge $\{x, x'\} \in E(G \setminus u)$; if all edges of $G \setminus u$ had at least two labels under L', then it would be $c(L') \geq 2(k-1) = 2k-2$. Let l_{unique} be the unique label of the edge $\{x, x'\}$. Also, without loss of generality, assume that x is further from u than x' is, i.e., the unique path from u to x goes through x'. For L to enable a (u, x)-journey in $G(L)$, it must assign to the edge $\{u, u'\}$ (at least) one label l that is strictly smaller than l_{unique}. Also, to enable a (x, u)-journey, L must assign to the edge $\{u, u'\}$ (at least) one label l' that is strictly greater than l_{unique} and, thus, different from label l (cf Figure 3). So, L assigns to $\{u, u'\}$ at least two labels, which makes the cost of L:

$$c(L) \geq c(L') + 2 = 2k - 3 + 2 = 2k - 1$$

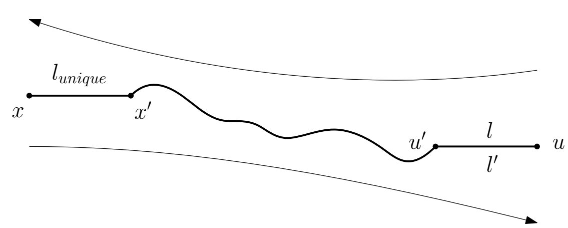

Fig. 3. L must assign to $\{u, u'\}$ at least 2 labels.

Therefore, in any case, for L to satisfy TC on G, it needs to have cost $c(L) \geq 2k - 1$.

(b) Consider a fixed, but arbitrary, spanning tree T of G and let a node, w, of degree 1 be the root of T. Also let w' be the single child of w in T and denote by T' the subtree of T that is rooted at w'. Let r be the length of the longest path from w' to any leaf of T', i.e., r is the radius of T'. We assign labels to the edges of T as follows:

Going upwards. Any edge of T' incident to a leaf gets label 1. Any edge $e = \{u, v\}$ of T', with $d(w', v) = d(w', u) + 1$, where the subtree T^* rooted at v has been labelled going upwards towards w', gets a label $l_e = max\{\text{all labels in } T^*\} + 1$ (cf. Figure 4).

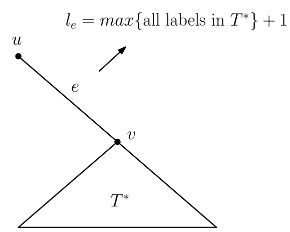

Fig. 4. Labelling "going upwards" to the root

The edge $\{w, w'\}$ We label the edge $\{w, w'\}$ of T with the single label $r+1$.

Going downwards. Any edge of T' incident to w' gets a label $r + 2$. Any edge e of T' in a path from w' to a leaf of T', the *parent edge*[6] of which has been labelled, going downwards, with label l', gets a label $l_e = l' + 1$.

We can easily implement the above process by topologically ordering the vertices of T in levels using *Breadth First Search* and implement the "going upwards" and "going downwards" procedures accordingly. The above method results in a labelling where:

1. each edge of T has 2 labels, except for the edge $\{w, w'\}$, which has a single label,
2. each edge of $E \setminus T$ has no label and
3. for each ordered pair of vertices $u, v \in V,\ u \neq v$, there is a (u, v)-journey.

To show 3, just notice that one can go from any vertex $u \in V$ to any other vertex $v \in V$, with $u, v \neq w$, by going up in T from u to w' and then going down in T from w' to v via strictly increasing labels, by construction. Finally, to realize journeys from w to some $u \in V$, one can go down in T, using strictly increasing labels (starting with the label $r + 1$), and to realize journeys from some $u \in V$ to w, one can go up in T, using strictly increasing labels (ending with the label $r + 1$).

□

Example *Figure 5 shows an example of the procedure described above. Notice that journeys between all pairs of vertices exist in the resulting temporal graph.*

[6] The edge before it in the sequence of edges from the root w' to the respective leaf.

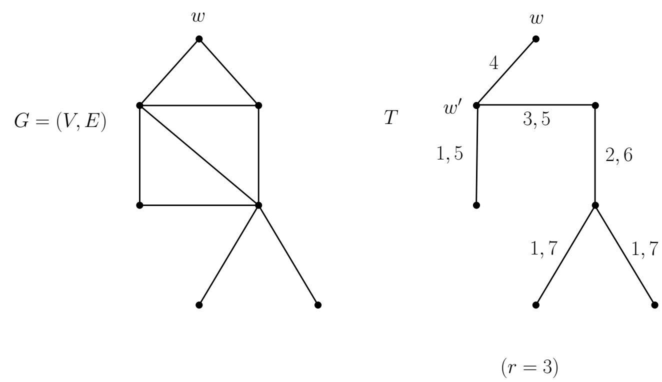

Fig. 5. Labelling a connected undirected graph to satisfy TC

Conjecture *We conjecture that for any connected undirected graph G of n vertices and for any labelling L that satisfies the property TC on G, the cost of L is $c(L) \geq 2n - 4$.*

Notice that the choice of $2n-4$ as the lower bound in the above conjecture is due to the fact that there are graphs, e.g., a cycle with $n = 4$ vertices, that can be made temporally connected using $2n - 4$ labels in total (cf. Figure 6); therefore, the lower bound $2n - 3$ which is shown for trees in Theorem 2(a) cannot be generic.

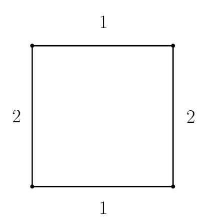

Fig. 6. Labelling a cycle of $n = 4$ vertices with $2n - 4 = 4$ labels to satisfy TC.

4 Minimal Temporal Designs

Suppose now that a temporal graph on a (strongly) connected (di)graph $G = (V, E)$ is *given* to a designer with the claim that it satisfies TC. In this scenario, the designer is allowed to only use the given set of edge availabilities, or a subset of them. If the given design is not minimal, they may wish to remove as many labels as possible, thus reducing the cost. Minimality of a design can be verified by running Algorithm 1 (cf. Section 2) for every $s \in V$.

4.1 A partial characterisation of minimal temporal graphs

As mentioned earlier, if many edges have the same label, we can encounter *trivial cases* of minimal temporal graphs. To avoid such cases, we focus our attention here to the class of SLSE temporal graphs, in which every edge only becomes available at one moment in time and no two different edges become available at the same time. Are there minimal SLSE temporal graphs with non linear (in the size of the graph) cost? For example, any complete SLSE temporal graph satisfies TC. Are all these $\Theta(n^2)$ labels needed for TC, i.e., are there minimal temporal complete graphs? As we prove in Theorem 4, the answer is negative. However, we give below a minimal temporal graph on n vertices with non-linear in n cost, namely with $O(n \log n)$ labels.

A minimal temporal design of $n \log n$ cost

Definition 10 (Hypercube graph). *The k-hypercube graph, commonly denoted Q_k, is a k-regular graph of 2^k vertices and $2^{k-1} \cdot k$ edges. The 1-hypercube is the graph of two vertices and one edge. Recursively, the n-hypercube is produced by taking two isomorphic copies of the $(n-1)$-hypercube and adding edges between the corresponding vertices.*

Definition 11 (Flat). *In geometry, a* flat *is a subset of the n-dimensional space that is congruent to a Euclidean space of lower dimension, e.g., the flats in the two-dimensional space are points and lines. In the n-dimensional space, there are flats of every dimension from 0, i.e., points, to $n-1$, i.e., hyperplanes.*

Theorem 3. *There exists an infinite class of minimal temporal graphs on n vertices with $\Theta(n \cdot \log n)$ edges and $\Theta(n \cdot \log n)$ labels, such that different edges have different labels.*

Proof. We present a minimal temporal graph on the hypercube graph of n vertices. Consider Protocol 2 for labelling the edges of $G = Q_k = (V, E)$. The temporal graph, $G(L)$, that this labelling procedure produces on the hypercube is minimal. Indeed, first we will prove that the temporal graph produced by Protocol 2 satisfies TC on $G = Q_k$.

Consider vertices $u, v \in V$ and the steps described in Protocol 3 to reach v, starting from u, via temporal edges. The procedure described in Protocol 3 gives a journey from u to v, which is also *unique*. It suffices to consider the k-bit binary representation of the vertices of G. Notice that if the hamming distance of the labels of two vertices $u, v \in V(G)$ is exactly m, then to reach v from u via a temporal path in the temporal graph on G, we need to move through vertices by consecutively swapping the bits in which u and v differ in the order of dimensions. This way, we maintain the strictly increasing order of the time labels we use and, swap by swap, we approach the destination. Note also that swapping only the bits in which u and v differ is the only way to not violate the increasing order of time labels we use: without loss of generality, suppose that the j^{th} bit of u is 1 and so is j^{th} bit of v. If, starting from u, we swap the j^{th}

bit to 0, i.e., we use an edge, e, on the j^{th} dimension, then at a future step, we again need to swap the j^{th} bit back to 1 (otherwise, we never reach v). However, the two swaps cannot be consecutive, because then we would use edge e twice and we violate the increasing order of labels. So, we would need to move to a higher dimension after the first of the two swaps; but, then, we have used labels that are larger than all the labels of the j^{th} dimension, so using any edge of the j^{th} dimension would also violate the increasing order of labels.

Protocol 2 Labelling the hypercube graph, $G = Q_k$

Consider the k dimensions of the hypercube $G = Q_k$, $x_1, x_2, \ldots, x_k$;
for $i = 1 \ldots k$ **do**
  Let $X_i := \{e_{i1}, e_{i2}, \ldots, e_{i2^{k-1}}\}$ be the list of edges in dimension x_i, in an arbitrary order;
  Let L_i be the (sorted from smallest to largest) list of labels $L_i := \{(i-1) \cdot 2^{k-1} + 1, (i-1) \cdot 2^{k-1} + 2, \ldots, i \cdot 2^{k-1}\}$;
for $i = 1 \ldots k$ **do**
  for $j = 1 \ldots 2^{k-1}$ **do**
    Assign the (current) first label of L_i to the (current) first edge of X_i ;
    Remove the (current) first label of L_i from the list;
    Remove the (current) first edge of X_i from the list;
return the produced temporal graph, $G(L)$;

Protocol 3 A temporal path from u to v in the temporal graph on $G = Q_k$

Input: The considered temporal graph on the hypercube $G = Q_k$, vertices $u, v \in V(G)$
Output: Array x of vertices, which the (u, v)-journey passes through
  $x[0] := u$;
  Find the flat of the smallest dimension, m, which both u and v lie on;
  Consider the increasing order of the m dimensions in that flat: $d[1], d[2], \ldots, d[m]$;
  for $i = 1 \ldots m$ **do**
    Use the incident edge of $x[i-1]$ that lies on dimension $d[i]$ and let $x[i]$ be the other endpoint of that edge;

Since our labelling gives a *unique* (u, v)-journey, for every $u, v \in V$, and since all labels assigned to the edges of E are used in the union of all those journeys, the deletion of any single label will violate TC. Therefore, $G(L)$ is minimal. Finally, note that the temporal graph $G(L)$ on the hypercube graph $G = Q_k$ has $n = 2^k$ vertices, $\frac{1}{2}n \cdot \log n$ edges and $\frac{1}{2}n \cdot \log n$ labels. □

A minimal temporal design of linear in n cost In the previous section, we showed that there are graphs of non-linear cost (in the number of vertices) that are minimal. Here, we show that there are classes of minimal graphs whose cost is linear in the number of their vertices.

Indeed, as seen in Lemma 1 (Section 3), the star graph of n vertices needs at least $\Theta(n)$ labels to satisfy TC and, in fact, we present there a TC satisfying labelling of $\Theta(n)$ labels (cf. Figure 1). Theorem 2(b) (Section 3) also gives a class of minimal temporal graphs of linear cost in the number of vertices. Therefore, we have the following Corollary:

Corollary 2. *There exists an infinite class of minimal temporal graphs on n vertices with $\Theta(n)$ edges and $\Theta(n)$ labels.*

SLSE Cliques of at least 4 vertices are not minimal The complete graph on n vertices, K_n, with an SLSE labelling L, i.e., a labelling that assigns a single label per edge, different labels to different edges, is an interesting case, since $K_n(L)$ always satisfies TC. However, it is not minimal as the theorem below shows.

Theorem 4. *Let $n \in \mathbb{N}$, $n \geq 4$ and denote by K_n the complete graph on n vertices. There exists* no minimal *SLSE temporal graph on $K_n(L)$. In fact, we can remove (at least) $\lfloor \frac{n}{4} \rfloor$ labels from any SLSE labelling on $K_n(L)$ without violating TC.*

Proof. The proof is divided in two parts, as follows:

(a) We first show that any SLSE labelling on the complete graph on 4 vertices produces a temporal graph that is not minimal, i.e., the theorem holds for K_4. Consider the six different labels a, b, c, d, x, y assigned by an SLSE labelling to the edges of K_4 as shown in Figure 7.

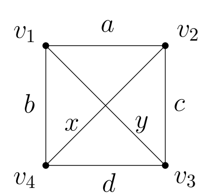

Fig. 7. Any SLSE labelling on K_4 is not minimal.

Up to their renaming, there are three possible cases for the labels a, b, c, d. Counting all the cases of *alternation*, *cycle*, and *entanglement* (see below) would give us all possible $4! = 24$ cases.

1. (Alternation) $a < b > d < c > a$.
   It is easy to see that in this case, both diagonals can be removed: v_1 can reach v_3 using labels a and then c; v_3 can reach v_1 using labels d and then b; v_2 can reach v_4 using labels a and then b; v_4 can reach v_2 using labels d and then c.

2. (Cycle) $a < b < d < c$.
   Here, diagonal x can be removed: v_2 can reach v_4 using labels a and then b; v_4 can reach v_2 using labels d and then c.
3. (Entanglement) $a < b < c < d$.
   This is a more complex case, for which we distinguish the following five sub-cases:
   i) $x < b$ and $y < c$.
   We can remove label a: v_1 can reach v_2 using labels y and then c; v_2 can reach v_1 using labels x and then b.
   ii) $x < b$ and $y > c$.
   We can remove label b: v_1 can reach v_4 using labels a, then c and then d; v_4 can reach v_1 using labels x, then c and then y (notice that $x < b < c < y$).
   iii) $x > b$ and $y > c$.
   We can remove label a: v_1 can reach v_2 using labels b and then x; v_2 can reach v_1 using labels c and then y.
   iv) $x > b$ and $b < y < c$.
   We can remove label x: v_2 can reach v_4 using labels a and then b; v_4 can reach v_2 using labels b, then y and then c.
   v) $x > b$ and $y < b$.
   We can remove label c: v_2 can reach v_3 using labels a, then b and then d; v_3 can reach v_2 using labels y, then b and then x.

   Notice that the coverage of the above five cases is complete (cf. Figure 8).

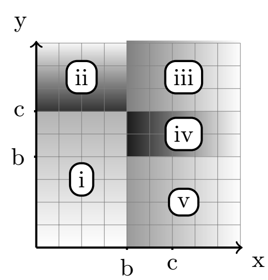

Fig. 8. The six sub-cases cover all possible scenarios of "entanglement".

(b) Now, consider the complete graph on $n \geq 4$ vertices, $K_n = (V, E)$. Partition V arbitrarily into $\lceil \frac{n}{4} \rceil$ subsets $V_1, V_2, \ldots, V_{\lceil \frac{n}{4} \rceil}$, such that $|V_i| = 4, \forall i = 1, 2, \ldots, \lceil \frac{n}{4} \rceil - 1$ and $|V_{\lceil \frac{n}{4} \rceil}| \leq 4$. In each 4-clique defined by V_i, $i = 1, 2, \ldots, \lfloor \frac{n}{4} \rfloor$, we can remove a "redundant" label, as shown in (a). The resulting temporal graph on K_n still preserves TC since for every ordered pair of vertices $u, v \in V$:

- if u, v are in the same V_i, $i = 1, 2, \ldots, \lfloor \frac{n}{4} \rfloor$, then there is a (u, v)-journey that uses time edges within the 4-clique on V_i, as proven in ((a)).

- if $u \in V_i$ and $v \in V_j,\ i \neq j$, then there is a (u, v)-journey that uses the (direct) time edge on $\{u, v\}$.

□

4.2 Computing the removal profit is APX-hard

Note that it is straightforward to check in polynomial time whether a given L satisfies TC on a given (di)graph G, by just checking for every possible (ordered) pair (u, v) of vertices in G whether there is a (u, v)-journey in $G(L)$. Recall that the removal profit is the largest number of labels that can be removed from a temporally connected graph without destroying TC. We now show that it is hard to approximate the value of the removal profit arbitrarily well for an arbitrary graph, i.e., there exists no PTAS[7] for this problem, unless P=NP. In our hardness proof below, we consider undirected graphs.

We prove our hardness result by providing an approximation preserving polynomial reduction from a variant of the maximum satisfiability problem, namely from the *monotone Max-XOR(*3*)* problem. Consider a monotone XOR-boolean formula ϕ with variables $x_1, x_2, \ldots, x_n$, i.e., a boolean formula that is the conjunction of XOR-clauses of the form $(x_i \oplus x_j)$, where no variable is negated. The clause $\alpha = (x_i \oplus x_j)$ is XOR-satisfied by a truth assignment τ if and only if $x_i \neq x_j$ in τ. The number of clauses of ϕ that are XOR-satisfied in τ is denoted by $|\tau(\phi)|$. If every variable x_i appears in exactly r XOR-clauses in ϕ, then ϕ is called a *monotone XOR(r)* formula. The *monotone Max-XOR(r)* problem is, given a monotone XOR(r) formula ϕ, to compute a truth assignment τ of the variables $x_1, x_2, \ldots, x_n$ that XOR-satisfies the largest possible number of clauses, i.e., an assignment τ such that $|\tau(\phi)|$ is maximized. The monotone Max-XOR(3) problem essentially encodes the *Max-Cut* problem on 3-regular (i.e., cubic) graphs, which is known to be APX-hard [4].

Lemma 2. *[4] The monotone Max-XOR(*3*) problem is APX-hard.*

Now we provide our reduction from the monotone Max-XOR(3) problem to the problem of computing $r(G, L)$. Let ϕ be an arbitrary monotone XOR(3) formula with n variables $x_1, x_2, \ldots, x_n$ and m clauses. Since every variable x_i appears in ϕ in exactly 3 clauses, it follows that $m = \frac{3}{2}n$. We will construct from ϕ a graph $G_\phi = (V_\phi, E_\phi)$ and a labelling L_ϕ of G_ϕ.

A very high-level description of the construction is as follows. G_ϕ is composed of gadgets that represent the variables x_i of the formula ϕ. Each variable x_i is assigned a "source" vertex s^{x_i} and three "sink" vertices $t_1^{x_i}, t_2^{x_i}, t_3^{x_i}$ in G_ϕ; each of them corresponds to one of the three clauses of ϕ in which x_i appears. The gadgets of G_ϕ are connected in such a way that, if $(x_i \oplus x_j)$ is a clause of ϕ, then one of the sink vertices of x_i coincides with one of the sink vertices of x_j. Furthermore it turns out that, by the construction, a journey from each source vertex s^{x_i} to the corresponding sink vertices $t_1^{x_i}, t_2^{x_i}, t_3^{x_i}$ represents a truth

[7] PTAS stands for Polynomial-Time Approximation Scheme.

assignment of the variable x_i. Moreover, the number of clauses of ϕ that can be satisfied by a truth assignment corresponds bijectively to the number of time-labels that can be removed from G_ϕ without destroying TC. Thus an optimum solution of the monotone Max-XOR(3) problem on ϕ corresponds to an optimal removal profit in G_ϕ.

Now we present the detailed construction of G_ϕ from the formula ϕ. First we construct for every variable x_i, where $1 \leq i \leq n$, the gadget-graph $G_{\phi,i}$ together with a labelling $L_{\phi,i}$ of its edges, as illustrated in Figure 9. In this figure, the labels of every edge in $L_{\phi,i}$ are drawn next to the edge. We call the induced subgraph of $G_{\phi,i}$ on the 4 vertices $\{s^{x_i}, u_0^{x_i}, w_0^{x_i}, v_0^{x_i}\}$ the *base* of $G_{\phi,i}$. Moreover, for every $p \in \{1,2,3\}$, we call the induced subgraph of $G_{\phi,i}$ on the 4 vertices $\{t_p^{x_i}, u_p^{x_i}, w_p^{x_i}, v_p^{x_i}\}$ the *pth branch* of $G_{\phi,i}$. Finally, we call the edges $\{u_0^{x_i}, w_0^{x_i}\}$ and $\{w_0^{x_i}, v_0^{x_i}\}$ the *transition edges* of the base of $G_{\phi,i}$ and, for every $p \in \{1,2,3\}$, we call the edges $\{u_p^{x_i}, w_p^{x_i}\}$ and $\{w_p^{x_i}, v_p^{x_i}\}$ the *transition edges* of the pth branch of $G_{\phi,i}$. For every $p \in \{1,2,3\}$ we associate the pth appearance of the variable x_i with the pth branch of $G_{\phi,i}$.

We continue the construction of $G_{\phi,i}$ and $L_{\phi,i}$ as follows. First, we add an edge between any possible pair of vertices $w_p^{x_i}, w_q^{x_j}$, where $p, q \in \{0,1,2,3\}$ and $i, j \in \{1,2,\ldots,n\}$, and we assign to this new edge $e = \{w_p^{x_i}, w_q^{x_j}\}$ the unique label $L_\phi(e) = \{7\}$. The addition of the above described edges is not illustrated in Figure 9. Note here that we add this edge $\{w_p^{x_i}, w_q^{x_j}\}$ also in the case where $i = j$ (and $p \neq q$).

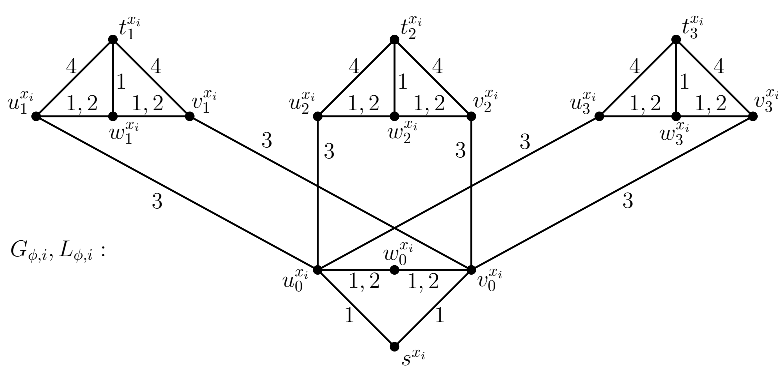

Fig. 9. The gadget $G_{\phi,i}$ for the variable x_i.

Intuitively, the base of $G_{\phi,i}$ (cf. Figure 9) corresponds to the variable x_i and, for every $p \in \{1,2,3\}$, the pth branch of $G_{\phi,i}$, together with the two edges $\{u_0^{x_i}, u_p^{x_i}\}$ and $\{v_0^{x_i}, v_p^{x_i}\}$, correspond to the clause of ϕ in which x_i appears for the pth time in ϕ.

Consider now a clause $\alpha = (x_i \oplus x_j)$ of ϕ. Assume that the variable x_i (resp. x_j) of α corresponds to the pth (resp. to the qth) appearance of x_i (resp. of x_j) in ϕ. Then we identify the vertices $u_p^{x_i}, v_p^{x_i}, w_p^{x_i}, t_p^{x_i}$ of the pth branch of $G_{\phi,i}$ with the vertices $v_q^{x_i}, u_q^{x_i}, w_q^{x_i}, t_q^{x_i}$ of the qth branch of $G_{\phi,j}$, respectively (cf. Figure 10(b)). Now we add an edge between any possible pair of vertices $t_p^{x_i}, t_q^{x_j}$, $i, j \in \{1, 2, \ldots, n\}$, and $p, q \in \{1, 2, 3\}$. We assign to this new edge $e = \{t_p^{x_i}, t_q^{x_j}\}$ the unique label $L_\phi(e) = \{7\}$.

Furthermore, for every $i \in \{1, 2, \ldots, n\}$ and every $p \in \{1, 2, 3\}$ we define for simplicity of notation the temporal paths $P_{i,p} = (s^{x_i}, u_0^{x_i}, u_p^{x_i}, t_p^{x_i})$ and $Q_{i,p} = (s^{x_i}, v_0^{x_i}, v_p^{x_i}, t_p^{x_i})$.

The intuition behind the composition of the gadget-graphs $G_{\phi,i}$ (cf. Figure 10(b)) is the following. If variable x_i is false in a truth assignment τ of ϕ, then all edges of the paths $P_{i,1}, P_{i,2}, P_{i,3}$ keep their labels as in L_ϕ. Otherwise, if x_i is true in τ, then all edges of the paths $Q_{i,1}, Q_{i,2}, Q_{i,3}$ keep their labels as in L_ϕ. Furthermore, depending on the value of x_i in the assignment τ, each of the transition edges $\{u_p^{x_i}, w_p^{x_i}\}$ and $\{w_p^{x_i}, v_p^{x_i}\}$, where $p \in \{1, 2, 3\}$, keeps exactly one of its two labels from L_ϕ. Consider now a clause $\alpha = (x_i \oplus x_j)$ of ϕ which corresponds to the pth branch of $G_{\phi,i}$ and to the qth branch of $G_{\phi,j}$. Then the only case where *both* edges $\{t_p^{x_i}, u_p^{x_i}\}$ and $\{t_p^{x_i}, v_p^{x_i}\}$ keep their labels from L_ϕ, is when the two variables x_i, x_j have *equal* truth value in the corresponding truth assignment τ of ϕ; that is, when the clause $\alpha = (x_i \oplus x_j)$ is *not* XOR-satisfied by τ. Therefore, intuitively, by a careful counting of the labels it turns out that, if more clauses can be satisfied by a truth assignment τ, then a TC preserving sub-labelling L of L_ϕ can be constructed which avoids more labels from L_ϕ, and vice versa (cf. Theorem 5).

To finalize the construction of the graph G_ϕ, we add a new vertex t_0 to ensure the existence of a temporal path between each pair of vertices of G_ϕ, as follows. This new vertex t_0 is adjacent to vertex $w_0^{x_n}$ and to all vertices in the set $\{s^{x_i}, t_1^{x_i}, t_2^{x_i}, t_3^{x_i}, u_p^{x_i}, v_p^{x_i} : 1 \leq i \leq n,\ 0 \leq p \leq 3\}$. First we assign to the edge $\{t_0, w_0^{x_n}\}$ the unique label $L_\phi(\{t_0, w_0^{x_n}\}) = \{5\}$. Furthermore, for every vertex $t_p^{x_i}$, where $1 \leq i \leq n$ and $1 \leq p \leq 3$, we assign to the edge $\{t_0, t_p^{x_i}\}$ the unique label $L_\phi(\{t_0, t_p^{x_i}\}) = \{5\}$. Finally, for each of the vertices $z \in \{s^{x_i}, u_p^{x_i}, v_p^{x_i} : 1 \leq i \leq n,\ 0 \leq p \leq 3\}$ we assign to the edge $\{t_0, z\}$ the unique label $L_\phi(\{t_0, z\}) = \{6\}$. The addition of the vertex t_0 and the labels of the (dashed) edges incident to t_0 are illustrated in Figure 10(a). Denote the vertex sets $A = \{s^{x_i}, u_p^{x_i}, v_p^{x_i} : 1 \leq i \leq n,\ 0 \leq p \leq 3\}$, $B = \{w_p^{x_i} : 1 \leq i \leq n,\ 0 \leq p \leq 3\}$, and $C = \{t_p^{x_i} : 1 \leq i \leq n,\ 1 \leq p \leq 3\}$. Note that $V_\phi = A \cup B \cup C \cup \{t_0\}$. This completes the construction of the graph G_ϕ and its labelling L_ϕ.

For every $i \in \{1, 2, \ldots, n\}$ the graph $G_{\phi,i}$ has 16 vertices. Furthermore, for every $p \in \{1, 2, 3\}$, the 4 vertices of the pth branch of $G_{\phi,i}$ also belong to a branch of $G_{\phi,j}$, for some $j \neq i$. Therefore, together with the vertex t_0, the graph G_ϕ has in total $10n+1$ vertices. We now present the auxiliary lemmas 3-5 which are necessary for the proof of Theorem 5.

Lemma 3. *The labelling L_ϕ assigns $\frac{17}{4}n^2 + 28n + 1$ labels to the edges of G_ϕ.*

Proof. The vertex t_0 has in total 3 incident edges (to vertices $s^{x_i}, u_0^{x_i}, v_0^{x_i}$) to every base of a variable x_i of ϕ, 3 incident edges (to vertices $t_p^{x_i}, u_p^{x_i}, v_p^{x_i}$, where $1 \leq p \leq 3$) to every clause $(x_i \oplus x_j)$ of ϕ (i.e., to one branch of x_i and one branch of x_j), and one incident edge to vertex $w_0^{x_n}$. That is, t_0 has in total

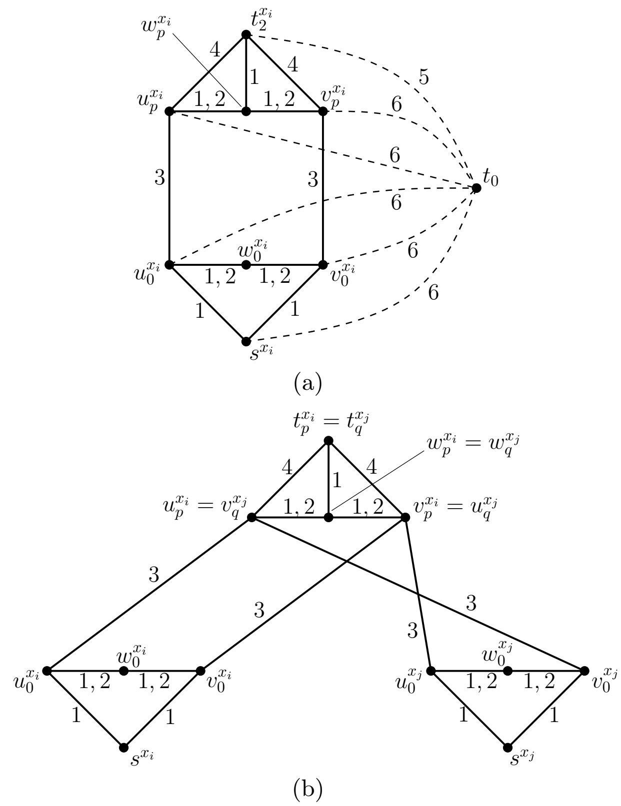

Fig. 10. (a) The addition of vertex t_0. There exists in G_ϕ also the edge $\{t_0, w_0^{x_n}\}$ with label $L_\phi(\{t_0, w_0^{x_n}\}) = \{5\}$. (b) The gadget for the clause $(x_i \oplus x_j)$.

$3n + 3m + 1 = 3n + 3 \cdot \frac{3}{2}n + 1 = \frac{15}{2}n + 1$ incident edges, each of them having one label in L_ϕ.

Furthermore there exist in total $\frac{m(m-1)}{2}$ edges among the vertices $\{t_p^{x_i} : 1 \leq i \leq n,\ 1 \leq p \leq 3\}$, as well as $\frac{(n+m)(n+m-1)}{2}$ edges among the vertices $\{w_p^{x_i} : 1 \leq i \leq n,\ 0 \leq p \leq 3\}$, each of them having one label in L_ϕ. Therefore, since $m = \frac{3}{2}n$, L_ϕ assigns in total $\frac{17}{4}n^2 - 2n$ labels for these edges.

Moreover, the labelling L_ϕ assigns to every variable x_i of ϕ in total 12 labels, i.e., two labels for each of the transition edges $\{u_0^{x_i}, w_0^{x_i}\}$, $\{w_0^{x_i}, v_0^{x_i}\}$ and one label for each of the edges $\{\{s^{x_i}, u_0^{x_i}\}, \{s^{x_i}, v_0^{x_i}\}, \{u_0^{x_i}, u_p^{x_i}\}, \{v_0^{x_i}, v_p^{x_i}\} : 1 \leq p \leq 3\}$.

Finally, L_ϕ assigns to every clause $(x_i \oplus x_j)$ of ϕ in total 7 labels, i.e., two labels for each of the transition edges $\{u_p^{x_i}, w_p^{x_i}\}$, $\{w_p^{x_i}, v_p^{x_i}\}$ and one label for each of the edges $\{u_p^{x_i}, t_p^{x_i}\}$, $\{v_p^{x_i}, t_p^{x_i}\}$, $\{t_p^{x_i}, w_p^{x_i}\}$, where x_i is associated with the pth branch of $G_{\phi,i}$. That is, L_ϕ assigns in total $7m = \frac{21}{2}n$ labels for all clauses of ϕ.

Summarizing, the labelling L_ϕ assigns to the edges of the graph G_ϕ a total of $\left(\frac{15}{2}n+1\right)+\left(\frac{17}{4}n^2-2n\right)+12n+\frac{21}{2}n=\frac{17}{4}n^2+28n+1$ labels. $\square$

Lemma 4. *The labelling L_ϕ satisfies TC on G_ϕ.*

Proof. We will prove that there exists a temporal path in L_ϕ between any pair of vertices of $V_\phi = A \cup B \cup C \cup \{t_0\}$.

For any two vertices $b, b' \in B$ there exists a temporal path from b to b' and from b' to b, due to the edge $\{b, b'\}$ with label 7. Similarly, for any two vertices $c, c' \in C$ there exists a temporal path from c to c' and from c' to c, due to the edge $\{c, c'\}$ with label 7. Let $a_1, a_2 \in A$. There exists a temporal path from a_1 to a_2 as follows: start from a_1, follow $P_{i,p}$ (or $Q_{i,p}$) upwards until $t_p^{x_i}$ with greatest label 4, then go to t_0 with label 5, and finally from t_0 to a_2 with label 6. In the special case where a_1 and a_2 lie on the same path $P_{i,p}$ (resp. $Q_{i,p}$) and a_1 appears before a_2 in $P_{i,p}$ (resp. $Q_{i,p}$), there exists clearly a temporal path from a_1 to a_2 along $P_{i,p}$ (resp. $Q_{i,p}$).

Let $a \in A$ and $b \in B$. Note that $b = w_p^{x_i}$ for some $i \in \{1, 2, \ldots, n\}$ and some $p \in \{0, 1, 2, 3\}$. There exists the temporal path from b to a as follows. First follow the edge $\{w_p^{x_i}, u_p^{x_i}\}$ (with label 1), then follow upwards the path $P_{i,p}$ until one of the vertices $\{t_1^{x_i}, t_2^{x_i}, t_3^{x_i}\}$ (with maximum label 4), then go to t_0 with label 5 and finally reach a with label 6. Furthermore there exists the temporal path from a to b as follows. Assume first that $a = s^{x_i}$, for some $i \in \{1, 2, \ldots, n\}$. If $b = w_0^{x_i}$ then there exists the temporal path on the edges $\{s^{x_i}, u_0^{x_i}\}$ (with label 1) and $\{u_0^{x_i}, w_0^{x_i}\}$ (with label 2). If $b \neq w_0^{x_i}$ then there exists the temporal path from s^{x_i} to $w_0^{x_i}$ (with maximum label 2), followed by the edge $\{w_0^{x_i}, b\}$ (with label 7). Assume now that $a \neq s^{x_i}$, for every $i \in \{1, 2, \ldots, n\}$. That is, $a = u_p^{x_i}$ or $a = v_p^{x_i}$, for some $i \in \{1, 2, \ldots, n\}$ and some $p \in \{0, 1, 2, 3\}$. If $b = w_p^{x_i}$ then there exists the temporal path from a to b on the edge $\{a, b\}$ (with label 1). If $b \neq w_p^{x_i}$ then there exists the temporal path from a to b through the edges $\{a, w_p^{x_i}\}$ (with label 1) and $\{w_p^{x_i}, b\}$ (with label 7). That is, there exists a temporal path in L_ϕ between any $a \in A$ and any $b \in B$.

Let $b \in B$, i.e., $b = w_p^{x_i}$ for some $i \in \{1, 2, \ldots, n\}$ and some $p \in \{0, 1, 2, 3\}$. Then there exists a temporal path from b to every vertex $c \in C$ as follows. If $p = 0$ then start with the edge $\{w_0^{x_i}, u_0^{x_i}\}$ (of label 1), continue upwards with a temporal path (of maximum label 4) until $t_1^{x_i} \in C$ and then continue to any other vertex $c \in C$ with the edge $\{t_1^{x_i}, c\}$ (of label 7). If $p \in \{1, 2, 3\}$ then reach $t_p^{x_i} \in C$ with the edge $\{w_p^{x_i}, t_p^{x_i}\}$ (of label 1) and continue to any other vertex $c \in C$ with the edge $\{t_p^{x_i}, c\}$ (of label 7). That is, there exists a temporal path from any $b \in B$ to any vertex of the set C. Now let $c \in C$, i.e., $c = t_p^{x_i}$ for some $i \in \{1, 2, \ldots, n\}$ and some $p \in \{1, 2, 3\}$. Then there exists a temporal path from c to every vertex $b \in B$ as follows. First reach the vertex $w_p^{x_i} \in B$ with the edge $\{t_p^{x_i}, w_p^{x_i}\}$ (of label 1) and then continue to any other vertex $c \in C$ with the edge $\{w_p^{x_i}, c\}$ (of label 7). That is, there exists a temporal path in L_ϕ between any $b \in B$ and any $c \in C$.

Let $a \in A$, i.e., $a \in \{s^{x_i}, u_p^{x_i}, v_p^{x_i}\}$ for some $i \in \{1, 2, \ldots, n\}$ and some $p \in \{0, 1, 2, 3\}$. Then there exists at least one path from a upwards to a vertex $c \in$

$\{t_1^{x_i}, t_2^{x_i}, t_3^{x_i}\}$ (with maximum label 4). Once we have (temporally) reached c from a, we can (temporally) continue to any other $c' \in C$ through the edge $\{c, c'\}$ (of label 7). That is, there exists a temporal path from any $a \in A$ to any vertex of C. Now let $c \in C$, i.e., $c = t_p^{x_i}$ for some $i \in \{1, 2, \ldots, n\}$ and some $p \in \{1, 2, 3\}$. Then there exists a temporal path from c to every vertex $a \in A$ as follows. First reach the vertex t_0 with the edge $\{t_p^{x_i}, t_0\}$ (of label 5) and then continue to any vertex $a \in A$ with the edge $\{t_0, a\}$ (of label 6). That is, there exists a temporal path in L_ϕ between any $a \in A$ and any $c \in C$.

Finally, there exists a temporal path between t_0 and every vertex of $A \cup C \cup \{w_0^{x_n}\}$, since t_0 is a neighbour with all these vertices. Let $b \in B$, i.e., $b = w_p^{x_i}$ for some $i \in \{1, 2, \ldots, n\}$ and some $p \in \{0, 1, 2, 3\}$. Then there exists a temporal path from $w_p^{x_i}$ to t_0 with the edges $\{w_p^{x_i}, u_p^{x_i}\}$ (with label 1) and $\{u_p^{x_i}, t_0\}$ (with label 6). On the other hand, there exists a temporal path from t_0 to every vertex $b = w_p^{x_i} \in B$, as follows. First reach the vertex $w_0^{x_n}$ with the edge $\{t_0, w_0^{x_n}\}$ (of label 5) and then, if $b \neq w_0^{x_n}$, continue with the edge $\{w_0^{x_n}, b\}$ (of label 7). That is, there exists a temporal path in L_ϕ between t_0 and any vertex in $A \cup B \cup C$.

Summarizing, there exists a temporal path between any pair of vertices of $V_\phi = A \cup B \cup C \cup \{t_0\}$, i.e., the labelling L_ϕ satisfies TC on G_ϕ. □

Lemma 5. *Let $L \subseteq L_\phi$ be a labelling of the graph G_ϕ. If L satisfies TC on G_ϕ, then L contains:*

(a) at least one label for every transition edge $\{u_p^{x_i}, w_p^{x_i}\}$ and $\{w_p^{x_i}, v_p^{x_i}\}$, where $i \in \{1, 2, \ldots, n\}$ and $p \in \{0, 1, 2, 3\}$,
(b) the label of each edge $\{t_p^{x_i}, w_p^{x_i}\}$, where $i \in \{1, 2, \ldots, n\}$ and $p \in \{1, 2, 3\}$,
(c) the labels of all edges of G_ϕ among the vertices $\{t_p^{x_i} : 1 \leq i \leq n,\ 1 \leq p \leq 3\}$,
(d) the labels of all edges among the vertices $\{w_p^{x_i} : 1 \leq i \leq n,\ 0 \leq p \leq 3\}$,
(e) the label of each edge incident to t_0, and
(f) the labels of all edges of the path $P_{i,p}$ or the labels of all edges of the path $Q_{i,p}$, where $i \in \{1, 2, \ldots, n\}$ and $p \in \{1, 2, 3\}$.

Proof.

(a) First assume that L does not keep any time label on the transition edge $\{u_p^{x_i}, w_p^{x_i}\}$ (resp. $\{w_p^{x_i}, v_p^{x_i}\}$), where $i \in \{1, 2, \ldots, n\}$ and $p \in \{0, 1, 2, 3\}$. Then there does not exist in L any temporal path from $u_p^{x_i}$ (resp. $v_p^{x_i}$) to $w_p^{x_i}$, even if L maintains all other edge labels from L_ϕ. This is a contradiction. Therefore L keeps at least one label on the transition edge $\{u_p^{x_i}, w_p^{x_i}\}$ (resp. $\{w_p^{x_i}, v_p^{x_i}\}$).
(b) Now assume that L does not contain the label of some edge $\{t_p^{x_i}, w_p^{x_i}\}$, where $i \in \{1, 2, \ldots, n\}$ and $p \in \{1, 2, 3\}$. Then there does not exist in L any temporal path from $t_p^{x_i}$ to any vertex $w_q^{x_j} \in B$, even if L maintains all other edge labels from L_ϕ. This is a contradiction to the assumption that L satisfies TC on G_ϕ. Therefore L contains the label of each edge $\{t_p^{x_i}, w_p^{x_i}\}$, where $i \in \{1, 2, \ldots, n\}$ and $p \in \{1, 2, 3\}$.
(c) Consider two vertices $t_p^{x_i} \neq t_q^{x_j}$, $1 \leq i < j \leq n$, $1 \leq p, q \leq 3$. If L does not contain the label of the edge $\{t_p^{x_i}, t_q^{x_j}\}$, then there does not exist in L any

temporal path from $t_p^{x_i}$ to $t_q^{x_j}$, which is a contradiction. Therefore L contains the labels of all edges of G_ϕ among the vertices $\{t_p^{x_i} : 1 \leq i \leq n,\ 1 \leq p \leq 3\}$.

(d) Assume that L does not contain the label of the edge $\{w_p^{x_i}, w_q^{x_j}\}$, for some $i, j \in \{1, 2, \ldots, n\}$ and $p, q \in \{0, 1, 2, 3\}$. Then there does not exist in L any temporal path from $w_p^{x_i}$ to $w_q^{x_j}$, which is a contradiction. Therefore L contains the labels of all edges among the vertices $\{w_p^{x_i} : 1 \leq i \leq n,\ 0 \leq p \leq 3\}$.

(e) We now prove that L contains the label of each edge incident to t_0. Recall that the neighbours of t_0 in G_ϕ are exactly the vertices of the set $A \cup C \cup \{w_0^{x_n}\}$. Assume L does not have the label of the edge $e = \{t_0, w_0^{x_n}\}$. Then there exists no temporal path in L from t_0 to any vertex $w_p^{x_i} \in B$, even if L maintains all other edge labels from L_ϕ. This is a contradiction to the assumption that L satisfies TC on G_ϕ. Now assume that there exists a vertex $a \in A = \{s^{x_i}, u_p^{x_i}, v_p^{x_i} : 1 \leq i \leq n,\ 0 \leq p \leq 3\}$ such that L does not have the label of the edge $e = \{t_0, a\}$. Then there does not exist in L any temporal path from vertex t_0 to vertex a, which is again a contradiction. Finally assume that there exists a vertex $t_p^{x_i} \in C$, such that L does not have the label of the edge $e = \{t_0, t_p^{x_i}\}$. Then there does not exist in L any temporal path from vertex $u_p^{x_i}$ to vertex s^{x_i}, which is a contradiction. Therefore L contains the label of each edge incident to t_0.

(f) Assume that L misses from L_ϕ at least one label of the path $P_{i,p}$ and at least one label of the path $Q_{i,p}$, for some $i \in \{1, 2, \ldots, n\}$ and $p \in \{1, 2, 3\}$. Then there does not exist any temporal path from s^{x_i} to $t_p^{x_i}$, which is a contradiction. Therefore L contains the labels of all edges of the path $P_{i,p}$ or the labels of all edges of the path $Q_{i,p}$, where $i \in \{1, 2, \ldots, n\}$ and $p \in \{1, 2, 3\}$.

□

We are now ready to provide the proof of Theorem 5.

Theorem 5. *There exists a truth assignment τ of ϕ with $|\tau(\phi)| \geq k$ if and only if there exists a TC satisfying labelling $L \subseteq L_\phi$ of G_ϕ such that $|L_\phi \setminus L| \geq 9n + k$.*

Proof. ($\Rightarrow$) Assume that there is a truth assignment τ that XOR-satisfies k clauses of ϕ. We construct a labelling L of G_ϕ by removing $9n + k$ labels from L_ϕ, as follows. First we keep in L all labels of L_ϕ on the edges incident to t_0. Furthermore we keep in L the label $\{7\}$ of all the edges $\{t_p^{x_i}, t_q^{x_j}\}$ and the label $\{7\}$ of all the edges $w_p^{x_i} w_q^{x_j}$. Moreover we keep in L the label $\{1\}$ of all the edges $\{t_p^{x_i}, w_p^{x_i}\}$. Let now $i = 1, 2, \ldots, n$. If $x_i = 0$ in τ, we keep in L the labels of the edges of the paths $P_{i,1}, P_{i,2}, P_{i,3}$, as well as the label 1 of the edge $\{v_0^{x_i}, w_0^{x_i}\}$ and the label 2 of the edge $\{w_0^{x_i}, u_0^{x_i}\}$. Otherwise, if $x_i = 1$ in τ, we keep in L the labels of the edges of the paths $Q_{i,1}, Q_{i,2}, Q_{i,3}$, as well as the label 1 of the edge $\{u_0^{x_i}, w_0^{x_i}\}$ and the label 2 of the edge $\{w_0^{x_i}, v_0^{x_i}\}$.

We now continue the labelling L as follows. Consider an arbitrary clause $\alpha = (x_i \oplus x_j)$ of ϕ. Assume that the variable x_i (resp. x_j) of the clause α corresponds to the pth (resp. to the qth) appearance of variable x_i (resp. x_j)

in ϕ. Then, by the construction of G_ϕ, the pth branch of $G_{\phi,i}$ coincides with the qth branch of $G_{\phi,j}$, i.e., $u_p^{x_i} = v_q^{x_j}$, $v_p^{x_i} = u_q^{x_j}$, $w_p^{x_i} = w_q^{x_j}$, and $t_p^{x_i} = t_q^{x_j}$ (cf. Figure 10(b)). Let α be XOR-satisfied in τ, i.e., $x_i = \overline{x_j}$. If $x_i = \overline{x_j} = 0$ (i.e., $x_i = 0$ and $x_j = 1$) then we keep in L the label 1 of the edge $\{v_p^{x_i}, w_p^{x_i}\}$ and the label 2 of the edge $\{w_p^{x_i}, u_p^{x_i}\}$, cf. Figure 11(a). In the symmetric case, where $x_i = \overline{x_j} = 1$ (i.e., $x_i = 1$ and $x_j = 0$), we keep in L the label 1 of the edge $\{u_p^{x_i}, w_p^{x_i}\}$ and the label 2 of the edge $\{w_p^{x_i}, v_p^{x_i}\}$. Let now α be XOR-unsatisfied in τ, i.e., $x_i = x_j$. Then, in both cases where $x_i = x_j = 0$ and $x_i = x_j = 1$, we keep in L the label 1 of both edges $\{v_p^{x_i}, w_p^{x_i}\}$ and $\{w_p^{x_i}, u_p^{x_i}\}$, cf. Figure 11(b). This finalizes the labelling L of G_ϕ. It is easy to check that L satisfies TC on G_ϕ.

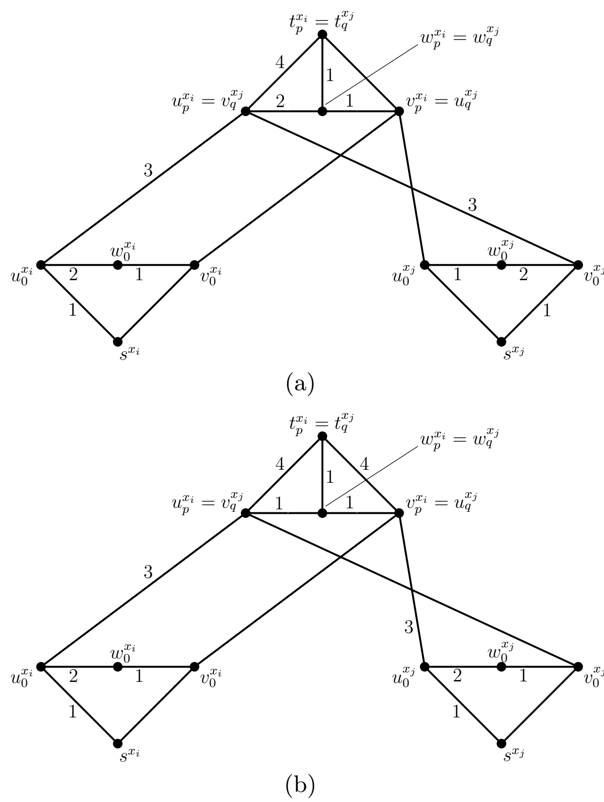

Fig. 11. The labelling $L \subseteq L_\phi$ of the edges of fig. 10(b) for the clause $\alpha = (x_i \oplus x_j)$ of ϕ, where (a) $x_i = \overline{x_j} = 0$ and (b) $x_i = x_j = 0$.

Summarizing, the labelling L misses in total 6 labels of L_ϕ for the edges $\{\{s^{x_i}, u_0^{x_i}\}, \{s^{x_i}, v_0^{x_i}\}, \{u_0^{x_i}, w_0^{x_i}\}, \{w_0^{x_i}, v_0^{x_i}\}, \{u_0^{x_i}, u_p^{x_i}\}, \{v_0^{x_i}, v_p^{x_i}\} : 1 \leq p \leq 3, i = 1, 2, \ldots, n\}$. That is, L misses in total $6n$ labels of

L_ϕ for all variables $x_1, x_2, \ldots, x_n$. For each of the k XOR-satisfied clauses $(x_i \oplus x_j)$ of ϕ, the labelling L misses in total 3 labels of L_ϕ for the edges $\{u_p^{x_i}, w_p^{x_i}\}$, $\{w_p^{x_i}, v_p^{x_i}\}$, $\{u_p^{x_i}, t_p^{x_i}\}$, $\{v_p^{x_i}, t_p^{x_i}\}$, $\{t_p^{x_i}, w_p^{x_i}\}$, where x_i is associated with the pth branch of $G_{\phi,i}$. That is, L misses in total $3k$ labels of L_ϕ for all XOR-satisfied clauses. Furthermore, for each of the $m-k$ XOR-satisfied clauses $(x_i \oplus x_j)$ of ϕ, the labelling L misses in total 2 labels of L_ϕ for the edges $\{u_p^{x_i}, w_p^{x_i}\}$, $\{w_p^{x_i}, v_p^{x_i}\}$, $\{u_p^{x_i}, t_p^{x_i}\}$, $\{v_p^{x_i}, t_p^{x_i}\}$, $\{t_p^{x_i}, w_p^{x_i}\}$, where x_i is associated with the pth branch of $G_{\phi,i}$. That is, L misses in total $2(m-k) = 3n-2k$ labels of L_ϕ for all XOR-satisfied clauses. All other labels of L_ϕ remain in the labelling $L \subseteq L_\phi$. Therefore, L misses in total $6n+3k+3n-2k = 9n+k$ labels from L_ϕ.

($\Leftarrow$) Assume that $r(G_\phi, L_\phi) \geq 9n+k$ and let $L \subseteq L_\phi$ be a TC preserving labelling of G_ϕ with $|L_\phi \setminus L| = r(G_\phi, L_\phi) \geq 9n+k$, i.e., $G_\phi(L)$ is minimal. Let $i \in \{1, 2, \ldots, n\}$. For every $p \in \{1, 2, 3\}$, L contains by Lemma 5(f) the labels of all edges of the path $P_{i,p}$ *or* the labels of all edges of the path $Q_{i,p}$. Therefore, there exist at least two indices $p_1, p_2 \in \{1, 2, 3\}$ such that L contains the labels of all edges of the paths P_{i,p_1}, P_{i,p_2} *or* the labels of all edges of the paths Q_{i,p_1}, Q_{i,p_2}. Without loss of generality let $p_1 = 1$ and $p_2 = 2$ and let L contain the labels of all edges of the paths $P_{i,1}, P_{i,2}$ (the other cases can be dealt with in the same way by symmetry). Assume that L also contains the labels of all edges of the path $Q_{i,3} = (s^{x_i}, v_0^{x_i}, v_3^{x_i}, t_3^{x_i})$. Then we can modify the labelling L to a labelling L' as follows. First remove from L the labels of the edges $\{s^{x_i}, v_0^{x_i}\}$ and $\{v_0^{x_i}, v_3^{x_i}\}$ and add instead the labels of the edges $\{u_0^{x_i}, u_3^{x_i}\}$ and $\{u_3^{x_i}, t_3^{x_i}\}$ (if they do not exist yet in L). Furthermore change the labels of the transition edges $\{v_0^{x_i}, w_0^{x_i}\}$ and $\{w_0^{x_i}, u_0^{x_i}\}$ to the labels 1 and 2, respectively. Note that in the resulting labelling L', both edges $\{u_3^{x_i}, t_3^{x_i}\}$ and $\{v_3^{x_i}, t_3^{x_i}\}$ are labelled. Furthermore $L' \subseteq L_\phi$ and L' does not have more labels than L, and thus $|L_\phi \setminus L'| \geq |L_\phi \setminus L| = r(G_\phi, L_\phi)$. Moreover, it is easy to check that L' still satisfies TC on G_ϕ, as L satisfies TC as well. So, it must also be $|L_\phi \setminus L'| = r(G_\phi, L_\phi)$, i.e., $G_\phi(L')$ is also minimal. Therefore, we may assume without loss of generality that for any minimal labelling $L \subseteq L_\phi$, L contains the labels of all edges of the paths $P_{i,1}, P_{i,2}, P_{i,3}$ *or* the labels of all edges of the paths $Q_{i,1}, Q_{i,2}, Q_{i,3}$.

From Lemma 5(a), L contains at least $2n+2m$ labels on the edges of the form $\{u_p^{x_i}, w_p^{x_i}\}$ or $\{w_p^{x_i}, v_p^{x_i}\}$, since there are exactly $2n$ transition edges on the different bases of G_ϕ and $2m$ transition edges on the different branches of G_ϕ. From Lemma 5(b), L contains m additional labels, one for each branch, more specifically for the respective edge $\{t_p^{x_i}, w_p^{x_i}\}$ of the branch. From Lemma 5(c), L contains $\frac{m(m-1)}{2}$ extra labels among the vertices $\{t_p^{x_i} : 1 \leq i \leq n, 1 \leq p \leq 3\}$. From Lemma 5(d), L also contains $\frac{(n+m)(n+m-1)}{2}$ additional labels among the vertices $\{w_p^{x_i} : 1 \leq i \leq n, 0 \leq p \leq 3\}$. From Lemma 5(e), L also contains $\frac{15}{2}n+1$ labels on the edges incident to t_0. Finally, from Lemma 5(f), L contains at least $4n+m$ additional labels: for each $G_{\phi,i}$, L contains at least 4 labels, namely one label on the base edge $\{s^{x_i}, u_0^{x_i}\}$ or on the base edge $\{s^{x_i}, v_0^{x_i}\}$ and, for every $p \in \{1, 2, 3\}$, one label on the edge $\{u_0^{x_i}, u_p^{x_i}\}$ or on the edge $\{v_0^{x_i}, v_p^{x_i}\}$; also, for each branch of G_ϕ, L contains at least 1 label, namely a label on the edge $\{u_p^{x_j}, t_p^{x_j}\}$ or on the edge $\{v_p^{x_j}, t_p^{x_j}\}$, for some $p \in \{1, 2, 3\}$ and $j \in \{1, 2, \ldots, m\}$.

Notice that all the labels of L mentioned above are on different edges, so no subset of labels has been accounted for more than once. Therefore, since $m = \frac{3n}{2}$, L contains at least:

$$c(L) \geq \frac{17}{4}n^2 + 17n + \frac{n}{2} + 1 \tag{1}$$

labels.

Now we construct from the labelling $L \subseteq L_\phi$ a truth assignment τ for the formula ϕ as follows. For every $i \in \{1, 2, \ldots, n\}$, if L contains the labels of all edges of the paths $P_{i,1}, P_{i,2}, P_{i,3}$, then we define $x_i = 0$ in τ. Otherwise, if L contains the labels of all edges of the paths $Q_{i,1}, Q_{i,2}, Q_{i,3}$, then we define $x_i = 1$ in τ. We will prove that $|\tau(\phi)| \geq k$, i.e., that τ XOR-satisfies at least k clauses of the formula ϕ.

Let $\alpha = (x_i \oplus x_j)$, where $i, j \in \{1, 2, \ldots, n\}$, be a clause of ϕ that is *not* XOR-satisfied by τ in ϕ. Let x_i (resp. x_j) be associated with the pth (resp. qth) branch of $G_{\phi,i}$ (resp. of $G_{\phi,j}$). Since α is *not* XOR-satisfied, either $x_i = x_j = 0$ or $x_i = x_j = 1$ in τ. If $x_i = x_j = 0$ in τ, it follows by the definition of the assignment τ that the labelling L contains the labels of all edges of the path $P_{i,p}$ *and* of the path $P_{j,q}$. Therefore, the p^{th} branch of $G_{\phi,i}$, which is identified with the q^{th} branch of $G_{\phi,j}$, has both edges $\{t_p^{x_i}, u_p^{x_i}\} \equiv \{t_q^{x_j}, v_q^{x_j}\}$ and $\{t_p^{x_i}, v_p^{x_i}\} \equiv \{t_q^{x_j}, u_q^{x_j}\}$ labelled under L, with one label each. The same holds if $x_i = x_j = 1$, where all edges of both paths $Q_{i,p}$ *and* $Q_{j,q}$ are labelled. So, for all the branches of G_ϕ that correspond to non-satisfied clauses of ϕ by the truth assignment τ, L contains an additional label (to the ones accounted for by using the result of Lemma 5(f)). The number of clauses that are not satisfied by τ in ϕ is exactly $m - |\tau(\phi)| = \frac{3}{2}n - |\tau(\phi)|$.

Thus, it follows by Equation (1), by adding the extra $\frac{3}{2}n - |\tau(\phi)|$, that L contains in total at least:

$$\begin{aligned} c(L) &\geq \frac{17}{4}n^2 + 17n + \frac{n}{2} + 1 + (\frac{3n}{2} - |\tau(\phi)|) \\ &= \frac{17}{4}n^2 + 19n + 1 - |\tau(\phi)| \end{aligned}$$

labels.

Recall now that we have already shown in Lemma 3 that L_ϕ has a total of $\frac{17}{4}n^2 + 28n + 1$ labels. Therefore, we have:

$$|L_\phi \setminus L| = c(L_\phi) - c(L) \leq 9n + |\tau(\phi)|.$$

However, by our initial assumption:

$$|L_\phi \setminus L| = r(G_\phi, L_\phi) \geq 9n + k.$$

Therefore $9n + k \leq |L_\phi \setminus L| \leq 9n + |\tau(\phi)|$, and thus $|\tau(\phi)| \geq k$, i.e., the truth assignment τ satisfies at least k clauses of ϕ. This completes the proof of the theorem. ☐

The next corollary follows immediately by Theorem 5.

Corollary 3. *Let $OPT_{mon\text{-}Max\text{-}XOR(3)}(\phi)$ the greatest number of clauses that can be simultaneously XOR-satisfied by a truth assignment of ϕ. Then $r(G_\phi, L_\phi) = 9n + OPT_{mon\text{-}Max\text{-}XOR(3)}(\phi)$.*

Proof. Let τ be a truth assignment that satisfies $k = \mathrm{OPT}_{\text{mon-Max-XOR(3)}}(\phi)$ clauses of ϕ. Then there exists by Theorem 5 a TC satisfying labelling $L \subseteq L_\phi$ of G_ϕ such that $|L_\phi \setminus L| \geq 9n + k$. Thus, since $r(G_\phi, L_\phi) \geq |L_\phi \setminus L|$, it follows that $r(G_\phi, L_\phi) \geq 9n + \mathrm{OPT}_{\text{mon-Max-XOR(3)}}(\phi)$. Conversely, let $L \subseteq L_\phi$ be a labelling of G_ϕ such that $|L_\phi \setminus L| = r(G_\phi, L_\phi)$. Then there exists by Theorem 5 a truth assignment τ that satisfies at least $r(G_\phi, L_\phi) - 9n$ clauses of ϕ. Thus $\mathrm{OPT}_{\text{mon-Max-XOR(3)}}(\phi) \geq r(G_\phi, L_\phi) - 9n$, which completes the proof. □

Using Theorem 5 and Corollary 3, we are now ready to prove the main theorem of this section.

Theorem 6. *The problem of computing $r(G, L)$ on an undirected temporally connected graph $G(L)$ is APX-hard.*

Proof. Denote by $\mathrm{OPT}_{\text{mon-Max-XOR(3)}}(\phi)$ the greatest number of clauses that can be simultaneously XOR-satisfied by a truth assignment of ϕ. The proof is done by an *L-reduction* [31] from the monotone Max-XOR(3) problem, i.e. by an approximation preserving reduction which linearly preserves approximability features. For such a reduction, it suffices to provide a polynomial-time computable function g and two constants $\alpha, \beta > 0$ such that:

- $r(G_\phi, L_\phi) \leq \alpha \cdot \mathrm{OPT}_{\text{mon-Max-XOR(3)}}(\phi)$, for any monotone XOR(3) formula ϕ, and
- for any TC satisfying labelling $L \subseteq L_\phi$ of G_ϕ, $g(L)$ is a truth assignment for ϕ and $\mathrm{OPT}_{\text{mon-Max-XOR(3)}}(\phi) - |g(L)| \leq \beta \cdot (r(G_\phi, L_\phi) - |L_\phi \setminus L|)$, where $|g(L)|$ is the number of clauses of ϕ that are satisfied by $g(L)$.

We will prove the first condition for $\alpha = 13$. Note that a random truth assignment XOR-satisfies each clause of ϕ with probability $\frac{1}{2}$, and thus there exists an assignment τ that XOR-satisfies at least $\frac{m}{2}$ clauses of ϕ. Therefore $\mathrm{OPT}_{\text{mon-Max-XOR(3)}}(\phi) \geq \frac{m}{2} = \frac{3}{4}n$, and thus $n \leq \frac{4}{3}\mathrm{OPT}_{\text{mon-Max-XOR(3)}}(\phi)$. Now Corollary 3 implies that:

$$\begin{aligned} r(G_\phi, L_\phi) &= 9n + \mathrm{OPT}_{\text{mon-Max-XOR(3)}}(\phi) \\ &\leq 9 \cdot \frac{4}{3}\mathrm{OPT}_{\text{mon-Max-XOR(3)}}(\phi) + \mathrm{OPT}_{\text{mon-Max-XOR(3)}}(\phi) \\ &= 13 \cdot \mathrm{OPT}_{\text{mon-Max-XOR(3)}}(\phi) \end{aligned} \tag{2}$$

To prove the second condition for $\beta = 1$, consider an arbitrary labelling $L \subseteq L_\phi$ of G_ϕ. As described in the ($\Leftarrow$)-part of the proof of Theorem 5, we construct in polynomial time a truth assignment $g(L) = \tau$ that satisfies at least $|L_\phi \setminus L| - 9n$ clauses of ϕ, i.e. $|g(L)| = |\tau(\phi)| \geq |L_\phi \setminus L| - 9n$. Then:

$$\begin{aligned} OPT_{\text{mon-Max-XOR(3)}}(\phi) - |g(L)| &\leq OPT_{\text{mon-Max-XOR(3)}}(\phi) - |L_\phi \setminus L| + 9n \\ &= r(G_\phi, L_\phi) - 9n - |L_\phi \setminus L| + 9n \\ &= r(G_\phi, L_\phi) - |L_\phi \setminus L| \end{aligned} \tag{3}$$

This completes the proof of the Theorem. □

Note *In fact, we have also shown (Theorem 5) that the problem of computing the removal profit is NP-hard in the strong sense, since all numbers used in the reduction are constant integers.*

Open Problem *Is there a polynomial-time constant factor approximation algorithm to compute $r(G, L)$?*

4.3 Temporally connected random labellings have high removal profit

In this section, we show that dense graphs with random labels have the property TC and have a very high removal profit asymptotically almost surely. More specifically, we consider the complete graph and the Erdös-Renyi model of random graphs, $G_{n,p}$ and we examine whether we can delete labels from such temporal graphs and continue preserving TC.

The (single-labelled) model of temporal graphs that we consider here is that of *uniform random temporal graphs* [1].

Definition 12. *[1] A* uniform random temporal graph *is a graph G on n vertices, $n \in \mathbb{N}$, each edge of which receives exactly one label uniformly at random from a set $\{1, 2, \ldots, \alpha\}$, $\alpha \in \mathbb{N}$ and the selection of the label of an edge is independent from the selection of the label of any other edge.*

High removal profit in the complete graph

Theorem 7. *In the uniform random temporal graph where the underlying graph G is the complete graph (clique) of n vertices and $\alpha \geq 4$, we can delete all but $\Theta(n \log n)$ labels without violating TC, with probability at least $1 - \frac{1}{n^2}$.*

Proof. First, note that any set $\{1, 2, \ldots, \alpha\}$ of α consecutive natural numbers can be partitioned into 4 disjoint almost equal subsets of consecutive numbers, A_1, A_2, A_3, A_4. Indeed, let $\alpha = 4k + v$, where $k \in \mathbb{N}$ and $v \in \{1, 2, 3, 4\}$.

For $v = 0$, we use $A_1 = \{1, \ldots, k\}, A_2 = \{k + 1, \ldots, 2k\}, A_3 = \{2k + 1, \ldots, 3k\}, A_4 = \{3k + 1, \ldots, 4k\}$.

For $v = 1$, we use $A_1 = \{1, \ldots, k\}, A_2 = \{k + 1, \ldots, 2k\}, A_3 = \{2k + 1, \ldots, 3k\}, A_4 = \{3k + 1, \ldots, 4k + 1\}$.

For $v = 2$, we use $A_1 = \{1, \ldots, k\}, A_2 = \{k + 1, \ldots, 2k\}, A_3 = \{2k + 1, \ldots, 3k + 1\}, A_4 = \{3k + 2, \ldots, 4k + 2\}$.

For $v = 3$, we use $A_1 = \{1, \ldots, k\}, A_2 = \{k + 1, \ldots, 2k + 1\}, A_3 = \{2k + 2, \ldots, 3k + 2\}, A_4 = \{3k + 3, \ldots, 4k + 3\}$.

In any of the above four cases, each particular edge of the clique K_n receives a single random label l, with:

$$Pr[l \in A_i] \geq \frac{k}{4k + 3}, \ \forall i = 1, 2, 3, 4$$

Since $k \geq 1$ (because $\alpha \geq 4$), we have $\frac{k}{4k+3} \geq \frac{1}{7}$. So, we get the following Lemma:

Lemma 6. *For each particular edge e of K_n and for the label l that it receives, it holds that $Prob[l \in A_i] \geq \frac{k}{4k+3}$, $\forall i = 1, 2, 3, 4$.*

Now, colour *green*(g), *yellow*(y), *blue*(b) and *red*(r) the edges that are assigned a label in A_1, A_2, A_3 and A_4 respectively.

Definition 13. *A temporal router (cf. Figure 12) of a clique $G = K_n = (V, E)$ is a subgraph $R = (V_R, E_R)$ of G, with $2\gamma \log n + 1$ vertices, γ being a constant such that $\gamma \geq 4 \cdot \frac{1}{\log_2 \frac{2500}{2499}}$, with the following properties (all logarithms are with base 2 here):*

a) V_R is the union of a particular vertex v_0 (called the centre *of R) and two equisized vertex sets V_{in} and V_{out}, each of $\gamma \log n$ vertices, and*
b) R is the induced subgraph of G formed from V_R (so it is a clique itself).

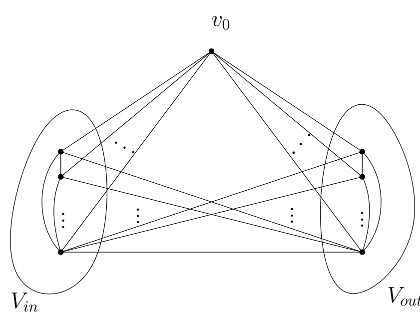

Fig. 12. Temporal router of a clique

Note that R has $|E_R| = 2\gamma \log n + \frac{(2\gamma \log n)\cdot(2\gamma \log n - 1)}{2}$ edges.

Let w, w' be any two vertices of the clique that are not in V_R. We consider the edges connecting w to V_{in} and the edges connecting w' to V_{out}; using those edges and the edges of R, there are $\gamma \log n$ edge-disjoint paths of length 4 (each) connecting w and w'. Let us call those paths *special paths* and note that every such path uses edges of the form $\{w, v_{in}\}, \{v_{in}, v_0\}, \{v_0, v_{out}\}, \{v_{out}, w'\}$, where $v_{in} \in V_{in}$ and $v_{out} \in V_{out}$ (cf. Figure 13).

Each special path $P = (w, v_{in}, v_0, v_{out}, w')$ connecting w and w' becomes a (w, w')-journey if the label l_1 of $\{w, v_{in}\}$ is in A_1, the label l_2 of $\{v_{in}, v_0\}$ is in A_2, the label l_3 of $\{v_0, v_{out}\}$ is in A_3, and the label l_4 of $\{v_{out}, w'\}$ is in A_4. Then, the probability that P is a journey is at least $\left(\frac{1}{7}\right)^4$, due to independence of the labels' selection.

Since all special paths that connect w and w' are edge-disjoint, the probability that none of them is a (w, w')-journey is:

$$Pr[\text{no special path is a } (w, w')\text{-journey}] = \left(1 - \frac{1}{7^4}\right)^{\gamma \log n} < \left(1 - \frac{1}{2500}\right)^{\gamma \log n} = n^{-\gamma \log \frac{2500}{2499}}.$$

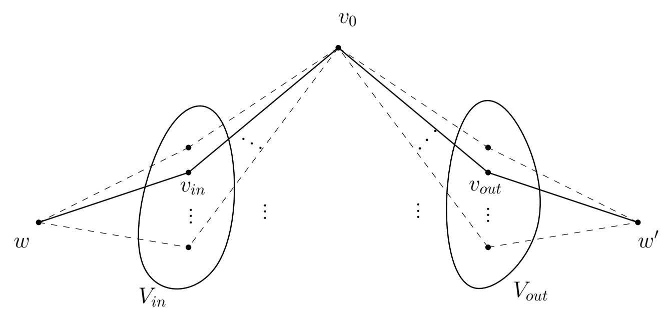

Fig. 13. A special path connecting w and w'.

Therefore, we have:

Lemma 7. *For any two particular vertices w, w' of $V \setminus V_R$, the probability that there is a special path P from w to w' that is a (w, w')-journey is at least $1 - n^{-\gamma \log_2 \frac{2500}{2499}}$.*

Now, we consider only the edges and labels of R and, for each $w \in V \setminus V_R$, we consider only the edges connecting w to each vertex of R; the sparsified graph $G' = (V, E')$ has, thus, $|E'| = \frac{(2\gamma \log n+1)\cdot 2\gamma \log n}{2} + \big(n - (2\gamma \log n + 1)\big) \cdot (2\gamma \log n + 1) = \Theta(n \log n + \log^2 n) = \Theta(n \log n)$ edges. We will show that we need only consider the edges (and labels) of G' to maintain TC in G, i.e., that G' itself is temporally connected, with probability at least $1 - \frac{1}{n^2}$.

Consider any pair, w, w', of vertices of the uniform random temporal graph on K_n and a temporal router R. Also, consider the graph G' as described above, with the labelling implied by the uniform random labelling on the clique. If $w, w' \in V_R$, then they are directly connected via a labelled edge in G' and thus a journey exists both ways between them. If $w \in V_R$ and $w \in V \setminus V_R$, then again there is a direct labelled edge in G' connecting w and w', so there is a journey between them either way.

It remains to examine the existence in G' of journeys between pairs of vertices $w, w',\ w \neq w'$, none of which is in V_R; there are at most n^2 such pairs of vertices. Under the random labelling on G, let $\mathcal{E}_1$ be the event that there exists a pair $w, w' \in V \setminus V_R$ such that there is no (w, w')-journey via a special path through R. Also, let $\mathcal{E}_2$ be the event that for a specific pair $w, w' \in V \setminus V_R$, there is no (w, w')-journey via a special path through R. Then,

$$Pr[\mathcal{E}_1] \leq n^2 Pr[\mathcal{E}_2] \text{ (by the Union Bound).}$$

So, we have:

$$Pr[G' \text{ is not temporally connected}] \leq n^2 n^{-\gamma \log_2 \frac{2500}{2499}}.$$

Note that Lemma 7 gives an upper bound on the probability of the event $\mathcal{E}_2$. Set γ to be $\gamma \geq 4 \cdot \frac{1}{\log_2 \frac{2500}{2499}}$. Then, we have:

$$Pr[G' \text{ is not temporally connected}] \leq n^{-2}.$$

□

High removal profit in dense random Erdös-Renyi graphs In this section, we consider the underlying graph $G = (V, E)$ to be an instance of the Erdös-Renyi graph model, $G_{n,p}$, with $n \geq 14$ and $p \geq 7\left(\frac{\gamma \ln n}{n}\right)^{\frac{1}{7}}$, $\gamma \geq 24$.

Definition 14 (Erdös-Renyi graphs). *An instance of $G_{n,p}$ is formed when for every pair of vertices u, v among a total number of n vertices, the edge $\{u, v\}$ is chosen to exist with probability p independently of any other edge.*

We will also use the Multiplicative Chernoff bound, as described below:

Fact (Chernoff bound [28]) *Suppose $X_1, \ldots, X_n$ are independent random variables taking values in $\{0, 1\}$. Let X denote their sum and let $\mu = E[X]$ denote the sum's expected value. Then, for $0 < \delta < 1$:*
$Pr[X > (1+\delta)\mu] \leq e^{-\frac{\delta^2 \mu}{3}}$*, and*
$Pr[X \leq (1-\delta)\mu] \leq e^{-\frac{\delta^2 \mu}{2}}$.

Notice that $G_{n,p}$ is almost surely connected for any $p \geq 2\frac{\ln n}{n}$ [8]. As in the previous section, we consider here a uniform random temporal graph on G, i.e., we consider each edge of G to receive exactly one label uniformly at random from a set $\{1, 2, \ldots, \alpha\}$, with $\alpha \geq 4$. The selection of the label of an edge is independent of the selection of the label of any other edge. Also, the label selection process is independent of the process of selection of edges in $G_{n,p}$. As in Theorem 7, we consider partitioning $\{1, 2, \ldots, \alpha\}$ into *four consecutive subsets,* A_1, A_2, A_3, A_4, *of consecutive positive integers*, where each subset is of size either $\lfloor \frac{\alpha}{4} \rfloor$ or $\lfloor \frac{\alpha}{4} \rfloor + 1$; such a partition is always possible. Now colour *green*(g), *yellow*(y), *blue*(b) and *red*(r) the edges that are assigned a label in A_1, A_2, A_3 and A_4, respectively. As in Lemma 6, we have:

Lemma 8. *For each particular edge of G and for the label l that it receives, it holds that $Prob[l \in A_i] \geq \frac{1}{7}$, $\forall i = 1, 2, 3, 4$.*

In such instances of $G_{n,p}$, we cannot assume the existence of cliques such as the clique of the temporal router used in the previous section. Indeed, even for very dense instances of $G_{n,p}$, with $p = \frac{1}{2}$, the largest clique is at most of size $2 \ln n$ [8].

In order to "sparsify" labelled instances G of $G_{n,p}$, by removing labels without violating TC, we need to guarantee the existence of much sparser routing subsets of G.

Definition 15. *Given two vertices v_1, v_2 of $G_{n,p}$, a* temporal router, $R(v_1, v_2)$, in an instance I of $G_{n,p}$ *is a subgraph of I that has vertices v_1, v_2 and additional vertices $a_1, \ldots, a_k$ and $b_1, \ldots, b_k$ so that:*

- *v_1 connects directly to each $a_i, b_i,\ i = 1, \ldots, k$,*
- *v_2 connects directly to each $a_i, b_i,\ i = 1, \ldots, k$,*
- *each pair a_i, b_i is directly connected, $i = 1, \ldots, k$,*
- *each edge $\{a_i, b_i\}$ receives a green label, $i = 1, \ldots, k$,*
- *each edge $\{a_i, v_1\}$ receives a yellow label, $i = 1, \ldots, k$,*
- *each edge $\{v_1, b_i\}$ receives a blue label, $i = 1, \ldots, k$,*
- *each edge $\{v_2, a_i\}$ receives a blue label, $i = 1, \ldots, k$,*
- *each edge $\{b_i, v_2\}$ receives a yellow label, $i = 1, \ldots, k$.*

Figures 14 and 15 show a temporal router $R(v_1, v_2)$ for $k = 1$ and $k = 2$ respectively.

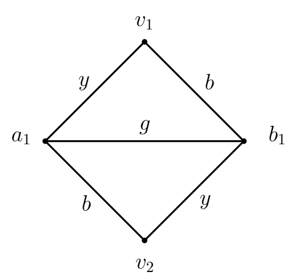

Fig. 14. Temporal router $R(v_1, v_2)$ for $k = 1$.

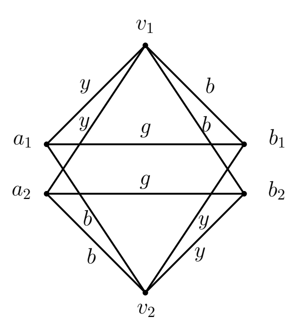

Fig. 15. Temporal router $R(v_1, v_2)$ for $k = 2$.

Note 1. A temporal router $R(v_1, v_2)$ in an instance I of $G_{n,p}$ is *temporally connected* since:

- any a_i can reach any b_j, via a journey through v_1, i.e., (a_i, v_1, b_j) is a journey,
- any b_i can reach any a_j, via a journey through v_2, i.e., (b_i, v_2, a_j) is a journey,

- any a_i can reach any $a_j \neq a_i$, via a journey through b_i and then v_2, i.e., (a_i, b_i, v_2, a_j) is a journey,
- any b_i can reach any $b_j \neq b_i$, via a journey through a_i and then v_1, i.e., (b_i, a_i, v_1, b_j) is a journey,
- v_1 can reach v_2, via any a_i, i.e., (v_1, a_i, v_2) is a journey,
- v_2 can reach v_1, via any b_i, i.e., (v_2, b_i, v_1) is a journey, and
- all other (temporal) connections are direct.

Definition 16. *We denote by R_i and call it the i^{th}* theta subgraph of $R(v_1, v_2)$ *the labelled subgraph of $R(v_1, v_2)$ induced by the vertices v_1, v_2, a_i, and b_i, for some $i = 1, \ldots, k$ (cf. Figure 16).*

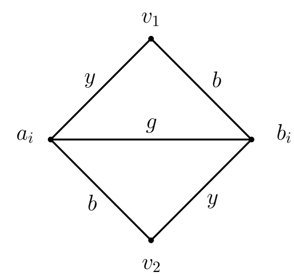

Fig. 16. The i^{th} theta subgraph, R_i.

Note that the following Lemma holds:

Lemma 9. *Let I be an instance of $G_{n,p}$ with a uniform random labelling from the set $\{1, 2, \ldots, \alpha\}$, $\alpha \geq 4$. Fix two vertices v_1, v_2 and $2k$ vertices a_i, b_i, $i = 1, \ldots, k$, in I. Then, for each particular $i = 1, \ldots, k$, the probability that the subgraph induced by the vertices v_1, v_2, a_i, b_i is a theta subgraph is:*

$$Pr[R_i \text{ exists in } I] \geq \left(\frac{p}{7}\right)^5$$

Proof. Each edge of R_i is realized in $G_{n,p}$ with probability p and receives the correct type of label (green, yellow, blue, or red) with probability at least $\frac{1}{7}$. Note also that the edges of different theta subgraphs R_i and R_j, $i \neq j$, are disjoint. Thus, in $G_{n,p}$, the random experiments of each of the theta subgraphs R_i appearing are *independent* from each other and each succeeds with probability at least $\left(\frac{p}{7}\right)^5$. □

Now, consider the set of vertices $V \setminus \{v_1, v_2\}$ and partition it into two almost equal sets V_1 and V_2; note that $|V_i| \geq \lfloor \frac{n}{2} \rfloor - 2 \geq \frac{n}{3} = n'$, $i = 1, 2$ (since $n \geq 14$). Consider a pairing of n' vertices of V_1 to n' vertices of V_2 and let the n' different pairs be the possible pairs of vertices a_i, b_i in a theta subgraph R_i. By Lemma 9 and since the random experiments are independent, the number of appearances of R_i is at least the number of successes in a Bernoulli distribution of n' trials, with success probability $\left(\frac{p}{7}\right)^5$ per trial. Therefore, by the Chernoff bound, we have the following Lemma:

Lemma 10. *Consider an instance I of a $G_{n,p}$ that has been labelled uniformly at random and fix two vertices v_1, v_2 in I. The probability that there is a temporal router $R(v_1, v_2)$ consisting of at least $k = \frac{n'}{2}\left(\frac{p}{7}\right)^5 = \frac{n}{6}\left(\frac{p}{7}\right)^5$ theta subgraphs R_i is at least $1 - e^{-\frac{1}{8}n'\left(\frac{p}{7}\right)^5} = 1 - e^{-\frac{n}{24}\left(\frac{p}{7}\right)^5}$.*

Corollary 4. *Note that, again by the Chernoff bound, k asymptotically almost surely* does not exceed $\frac{3}{2}n'\left(\frac{p}{7}\right)^5$, *since $Pr[k > \left(1 + \frac{1}{2}\right)E(k)] \leq e^{-\frac{n}{36}\left(\frac{p}{7}\right)^5}$.*

We now condition on the event, $\mathcal{E}_1$, that the instance I of a labelled $G_{n,p}$ has a temporal router $R(v_1, v_2)$ of at least $k = \frac{n}{6}\left(\frac{p}{7}\right)^5$ theta subgraphs. By Lemma 10, we know that:

$$Pr[\bar{\mathcal{E}_1}] \leq e^{-\frac{n}{24}\left(\frac{p}{7}\right)^5}.$$

Given that $R(v_1, v_2)$ exists, any vertex u that is not in $R(v_1, v_2)$ can reach any vertex u' that is also not in $R(v_1, v_2)$ *through* $R(v_1, v_2)$ via a journey, if u connects to some a_i directly with a green edge, and b_i connects to u' directly with a red edge. Then, (u, a_i, v_1, b_i, u') is a journey.

The probability of the edge $\{u, a_i\}$ being green *and* the edge $\{b_i, u'\}$ being red, for any i, is $\left(\frac{p}{7}\right)^2$ and it is independent of edge experiments inside $R(v_1, v_2)$. So, we have:

Lemma 11. *Condition on the event $\mathcal{E}_1$ of the existence of $R(v_1, v_2)$ in $G_{n,p}$ with at least $k = \frac{n}{6}\left(\frac{p}{7}\right)^5$ theta subgraphs. Let u, u' be any two (different) vertices of G that are not in $R(v_1, v_2)$. Then:*

$$Pr[\textit{there exists a } (u, u')\textit{-journey through } R(v_1, v_2)] \geq 1 - \left(1 - (\frac{p}{7})^2\right)^k$$

Proof. For the vertices u, u' as described above and any one of the k possible journeys of the form (u, a_i, v_1, b_i, u'), the probability that such a journey fails (i.e., is not realized) is at most $1 - \left(\frac{p}{7}\right)^2$. Therefore, given $\mathcal{E}_1$, we have:

$$Pr[\text{there exists no } (u, u')\text{-journey through } R(v_1, v_2)] \leq \left(1 - (\frac{p}{7})^2\right)^k$$

□

Let $\mathcal{E}_2$ be the event that given k pairs of vertices a_i, b_i in a possible $R(v_1, v_2)$, each vertex pair u, u', with $u \neq u'$ and $u, u' \notin V\big(R(v_1, v_2)\big)$, satisfies the following: there is at least one pair of vertices a_i, b_i such that u connects to a_i with a green edge *and* u' connects to b_i with a red edge.

Notice that $\bar{\mathcal{E}_2}$ is the event that there is a pair of vertices u, u' that are not in $R(v_1, v_2)$ that fails to connect as described above. Since the number of possible pairs of vertices u, u' is less than n^2, we have:

$$
\begin{aligned}
Pr[\bar{\mathcal{E}_2}] &\leq n^2\big(1-(\frac{p}{7})^2\big)^k \\
&\leq n^2 e^{-k(\frac{p}{7})^2} \\
&= e^{-k(\frac{p}{7})^2+2\ln n}
\end{aligned}
$$

Now, condition on $\mathcal{E}_1$ and on $\mathcal{E}_2$ (given $\mathcal{E}_1$). Then, for each vertex $u \notin V\big(R(v_1, v_2)\big)$, keep one of its green edges (to some a_i) and one of its red edges (to the corresponding b_i), since by $\mathcal{E}_2$, those exist. Then, remove all edges of I except for the edges of $R(v_1, v_2)$ and the two edges we keep for every vertex that is not in $R(v_1, v_2)$. Notice that the resulting labelled subgraph of I is temporally connected, since:

a) $R(v_1, v_2)$ is temporally connected itself, by construction,
b) any $u \notin V\big(R(v_1, v_2)\big)$ has a journey via $R(v_1, v_2)$ to any other $u' \in V$ in the graph,
c) any a_i or b_j can reach any $u \notin V\big(R(v_1, v_2)\big)$ via a journey through v_1(using first a green edge, if we start from a b_j vertex, and then using a yellow, a blue and a red edge to reach u), and
d) v_1 and v_2 can reach any $u \notin V\big(R(v_1, v_2)\big)$ via a journey through some vertex b_i (using first a blue -or yellow, respectively- edge to b_i, and then a red edge to u).

The temporally connected instance I of $G_{n,p}$ *after* the removal of redundant edges as described above has a number of labelled edges (i.e., time-edges) that is at most $2n + \Theta(k)$. Since $k = \frac{n}{6}\left(\frac{p}{7}\right)^5$, I has at most $\Theta(n + np^5)$ labels after the removal of the redundant edges.

Recall that we require $p \geq 7\left(\frac{\gamma \ln n}{n}\right)^{\frac{1}{7}}$, $\gamma \geq 24$. Therefore, we get the following:

Theorem 8. *Consider a $G_{n,p}$, with $p \geq 7\left(\frac{\gamma \ln n}{n}\right)^{\frac{1}{7}}$, for some $\gamma \geq 24$, labelled uniformly at random. Then, any instance I of $G_{n,p}$ needs only $\Theta(n+np^5)$ time-edges to be temporally connected, with probability at least $1 - 2e^{-\frac{\gamma}{24}\ln n}$.*

Proof. Any instance I of $G_{n,p}$ becomes temporally connected by using at most $\Theta(n+np^5)$ edges (and, thus, labels) as described above, with probability at least:

$$
Pr[\mathcal{E}_1] \cdot Pr[\mathcal{E}_2|\mathcal{E}_1] \geq \left(1 - e^{-\frac{n}{24}\left(\frac{p}{7}\right)^5}\right) \cdot \left(1 - e^{-k(\frac{p}{7})^2+2\ln n}\right). \tag{4}
$$

Since $p \geq 7\left(\frac{\gamma \ln n}{n}\right)^{\frac{1}{7}}$ and $k = \frac{n}{6}\left(\frac{p}{7}\right)^5$, we have:

$$
\begin{aligned}
k\left(\frac{p}{7}\right)^2 &= \frac{n}{6} \cdot \left(\frac{p}{7}\right)^7 \\
&\geq \frac{n}{6} \cdot \frac{\gamma \ln n}{n} \\
&= \frac{\gamma \ln n}{6}.
\end{aligned} \tag{5}
$$

Therefore, from relations (4) and (5), we have:

$$\begin{aligned} Pr[\mathcal{E}_1] \cdot Pr[\mathcal{E}_2|\mathcal{E}_1] &\geq \left(1 - e^{-\frac{n}{24}\left(\frac{p}{7}\right)^5}\right) \cdot \left(1 - e^{-\frac{\gamma \ln n}{6} + 2\ln n}\right) \\ &\geq \left(1 - e^{-\frac{n}{24}\left(\frac{p}{7}\right)^7}\right) \cdot \left(1 - e^{-\frac{\gamma \ln n}{6} + 2\ln n}\right) \end{aligned} \quad (6)$$

Again, since $p \geq 7\left(\frac{\gamma \ln n}{n}\right)^{\frac{1}{7}}$, we have that: $\left(\frac{p}{7}\right)^7 \geq \frac{\gamma \ln n}{n}$, so relation (6) becomes:

$$\begin{aligned} Pr[\mathcal{E}_1] \cdot Pr[\mathcal{E}_2|\mathcal{E}_1] &\geq \left(1 - e^{-\frac{n}{24}\frac{\gamma \ln n}{n}}\right) \cdot \left(1 - e^{-\frac{\gamma \ln n}{6} + 2\ln n}\right) \\ &\geq 1 - e^{-\frac{\gamma \ln n}{6} + 2\ln n} - e^{-\frac{n}{24}\cdot\frac{\gamma \ln n}{n}} \\ &= 1 - e^{-\ln n\left(\frac{\gamma}{6} - 2\right)} - e^{-\frac{\gamma}{24}\cdot \ln n} \\ &= 1 - n^{-\left(\frac{\gamma}{6} - 2\right)} - n^{-\frac{\gamma}{24}}. \end{aligned} \quad (7)$$

Now, since $\gamma \geq 24$, we have that $\frac{\gamma}{6} - 2 \geq \frac{\gamma}{24}$. Therefore, from relation (7), we get:

$$\begin{aligned} Pr[\mathcal{E}_1] \cdot Pr[\mathcal{E}_2|\mathcal{E}_1] &\geq 1 - 2n^{-\frac{\gamma}{24}} \\ &= 1 - 2e^{-\frac{\gamma}{24}\ln n}. \end{aligned}$$

□

Note that for the sparsest possible $G_{n,p}$ here, i.e., for $p = 7\left(\frac{\gamma \ln n}{n}\right)^{\frac{1}{7}}$, we need only $\Theta(n + n^{\frac{2}{7}}(\ln n)^{\frac{5}{7}}) = \Theta(n)$ edges (and, thus, labels) to satisfy TC, with probability at least $1 - 2e^{-\frac{\gamma}{24}\ln n}$, $\gamma \geq 24$.

5 Conclusions and further research

In this work, we study the complexity of testing and designing issues of nearly cost-optimal temporal networks that are temporally connected. It remains an open problem to provide a polynomial-time constant factor approximation algorithm for the computation of the removal profit in a given temporally connected temporal graph. Further research could also investigate the complexity of computing the removal profit in special classes of graphs, e.g., planar graphs or the grid. Extensions of this research also include the study of the interval temporal networks model, where edges can be available for continuous intervals of time, as well as a more in-depth study of models of random temporal networks.

References

1. Akrida, E.C., Gąsieniec, L., Mertzios, G.B., Spirakis, P.G.: Ephemeral networks with random availability of links: Diameter and connectivity. In: Proceedings of the 26th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA) (2014)

2. Akrida, E.C., Gąsieniec, L., Mertzios, G.B., Spirakis, P.G.: On temporally connected graphs of small cost. In: Proceedings of the 13th Workshop on Approximation and Online Algorithms (WAOA) (2015)
3. Akrida, E.C., Spirakis, P.G.: On verifying and maintaining connectivity of interval temporal networks. In: P. Bose, L.A. Gasieniec, K. Römer, R. Wattenhofer (eds.) Algorithms for Sensor Systems - 11th International Symposium on Algorithms and Experiments for Wireless Sensor Networks, ALGOSENSORS 2015, Patras, Greece, September 17-18, 2015, Revised Selected Papers, *Lecture Notes in Computer Science*, vol. 9536, pp. 142–154. Springer (2015). DOI 10.1007/978-3-319-28472-9_11. URL http://dx.doi.org/10.1007/978-3-319-28472-9_11
4. Alimonti, P., Kann, V.: Hardness of approximating problems on cubic graphs. In: Proceedings of the Third Italian Conference on Algorithms and Complexity, CIAC '97, pp. 288–298. Springer-Verlag, London, UK, UK (1997). URL http://dl.acm.org/citation.cfm?id=648256.752884
5. Angluin, D., Aspnes, J., Diamadi, Z., Fischer, M., Peralta, R.: Computation in networks of passively mobile finite-state sensors. Distributed Computing pp. 235–253 (2006)
6. Avin, C., Koucký, M., Lotker, Z.: How to explore a fast-changing world (cover time of a simple random walk on evolving graphs). In: Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP), pp. 121–132 (2008)
7. Barjon, M., Casteigts, A., Chaumette, S., Johnen, C., Neggaz, Y.M.: Testing temporal connectivity in sparse dynamic graphs. CoRR **abs/1404.7634** (2014). URL http://arxiv.org/abs/1404.7634
8. Bollobás, B.: Random Graphs, 2nd Edition. Cambridge University Press (2001)
9. Bui-Xuan, B.M., Ferreira, A., Jarry, A.: Computing shortest, fastest, and foremost journeys in dynamic networks. International Journal of Foundations of Computer Science **14**(2), 267–285 (2003)
10. Casteigts, A., Flocchini, P., Quattrociocchi, W., Santoro, N.: Time-varying graphs and dynamic networks. International Journal of Parallel, Emergent and Distributed Systems (IJPEDS) **27**(5), 387–408 (2012)
11. Chaintreau, A., Mtibaa, A., Massoulié, L., Diot, C.: The diameter of opportunistic mobile networks. In: J. Kurose, H. Schulzrinne (eds.) Proceedings of the 2007 ACM Conference on Emerging Network Experiment and Technology, CoNEXT 2007, New York, NY, USA, December 10-13, 2007, p. 12. ACM (2007). DOI 10.1145/1364654.1364670. URL http://doi.acm.org/10.1145/1364654.1364670
12. Clementi, A.E.F., Macci, C., Monti, A., Pasquale, F., Silvestri, R.: Flooding time of edge-markovian evolving graphs. SIAM Journal on Discrete Mathematics (SIDMA) **24**(4), 1694–1712 (2010)
13. Dutta, C., Pandurangan, G., Rajaraman, R., Sun, Z., Viola, E.: On the complexity of information spreading in dynamic networks. In: Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 717–736 (2013)
14. Fleischer, L., Skutella, M.: Quickest flows over time. SIAM J. Comput. **36**(6), 1600–1630 (2007). DOI 10.1137/S0097539703427215. URL http://dx.doi.org/10.1137/S0097539703427215
15. Fleischer, L., Tardos, É.: Efficient continuous-time dynamic network flow algorithms. Operations Research Letters **23**(3-5), 71–80 (1998)
16. Gavoille, C., Peleg, D., Perennes, S., Raz, R.: Distance labeling in graphs. In: Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 210–219 (2001)

17. Gupta, A., Krishnaswamy, R., Ravi, R.: Online and stochastic survivable network design. SIAM J. Comput. **41**(6), 1649–1672 (2012). DOI 10.1137/09076725X. URL http://dx.doi.org/10.1137/09076725X
18. Katz, M., Katz, N.A., Korman, A., Peleg, D.: Labeling schemes for flow and connectivity. SIAM Journal on Computing **34**(1), 23–40 (2004)
19. Kempe, D., Kleinberg, J.M., Kumar, A.: Connectivity and inference problems for temporal networks. In: Proceedings of the 32nd annual ACM symposium on Theory of computing (STOC), pp. 504–513 (2000)
20. Klinz, B., Woeginger, G.J.: One, two, three, many, or: complexity aspects of dynamic network flows with dedicated arcs. Oper. Res. Lett. **22**(4-5), 119–127 (1998). DOI 10.1016/S0167-6377(98)00009-1. URL http://dx.doi.org/10.1016/S0167-6377(98)00009-1
21. Koch, R., Nasrabadi, E., Skutella, M.: Continuous and discrete flows over time - A general model based on measure theory. Math. Meth. of OR **73**(3), 301–337 (2011). DOI 10.1007/s00186-011-0357-2. URL http://dx.doi.org/10.1007/s00186-011-0357-2
22. Kuhn, F., Lynch, N.A., Oshman, R.: Distributed computation in dynamic networks. In: Proceedings of the 42nd Annual ACM Symposium on Theory of Computing (STOC), pp. 513–522 (2010)
23. Lau, L.C., Naor, J., Salavatipour, M.R., Singh, M.: Survivable network design with degree or order constraints. SIAM J. Comput. **39**(3), 1062–1087 (2009). DOI 10.1137/070700620. URL http://dx.doi.org/10.1137/070700620
24. Lau, L.C., Singh, M.: Additive approximation for bounded degree survivable network design. SIAM J. Comput. **42**(6), 2217–2242 (2013). DOI 10.1137/110854461. URL http://dx.doi.org/10.1137/110854461
25. Mertzios, G.B., Michail, O., Chatzigiannakis, I., Spirakis, P.G.: Temporal network optimization subject to connectivity constraints. In: Proceedings of the 40th International Colloquium on Automata, Languages and Programming (ICALP), Part II, pp. 657–668 (2013)
26. Michail, O., Chatzigiannakis, I., Spirakis, P.G.: New Models for Population Protocols. Synthesis Lectures on Distributed Computing Theory. Morgan & Claypool Publishers (2011)
27. Michail, O., Chatzigiannakis, I., Spirakis, P.G.: Causality, influence, and computation in possibly disconnected synchronous dynamic networks. In: OPODIS, pp. 269–283 (2012)
28. Mitzenmacher, M., Upfal, E.: Probability and Computing: Randomized Algorithms and Probabilistic Analysis. Cambridge University Press (2005). URL https://books.google.co.uk/books?id=0bAYl6d7hvkC
29. Molloy, M., Reed, B.: Graph colouring and the probabilistic method, *Algorithms and Combinatorics*, vol. 23. Springer (2002)
30. O'Dell, R., Wattenhofer, R.: Information dissemination in highly dynamic graphs. In: Proceedings of the 2005 joint workshop on Foundations of Mobile Computing (DIALM-POMC), pp. 104–110 (2005)
31. Papadimitriou, C.H., Yannakakis, M.: Optimization, approximation, and complexity classes. J. Comput. Syst. Sci. **43**(3), 425–440 (1991). DOI 10.1016/0022-0000(91)90023-X. URL http://dx.doi.org/10.1016/0022-0000(91)90023-X
32. Scheideler, C.: Models and techniques for communication in dynamic networks. In: Proceedings of the 19th Annual Symposium on Theoretical Aspects of Computer Science (STACS), vol. 2285, pp. 27–49 (2002)

33. Whitbeck, J., Dias de Amorim, M., Conan, V., Guillaume, J.L.: Temporal reachability graphs. In: Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, Mobicom '12, pp. 377–388. ACM, New York, NY, USA (2012). DOI 10.1145/2348543.2348589. URL http://doi.acm.org/10.1145/2348543.2348589